DOCUMENT RESUME

ED 304 125 IR 013 698

AUTHOR	Ryan, William C., Ed.
TITLE	National Educational Computing Conference Proceedings (9th, Dallas, Texas, June 15-17, 1988).
INSTITUTION	National Educational Computing Conference.
REPORT NO	ISBN-0-924667-45-1
PUB DATE	Jun 88
NOTE	450p.; For the 1987 conference proceedings, see ED 291 346.
PUB TYPE	Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - Research/Technical (143)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; Computer Science Education; *Computer Software; Elementary Secondary Education; Higher Education; *Microcomputers; Staff Development; Teacher Education; *Telecommunications
IDENTIFIERS	*Computer Equity

ABSTRACT

The more than 200 papers and panel, project, and special session reports represented in this collection focus on innovations, trends, and research on the use of computers in a variety of educational settings. Of these, the full text is provided for 37 presentations and abstracts for 182. The topics discussed include: computer applications in mathematics, social studies, science, writing, language arts, reading, and art; teacher education; computer equity; computer-based education policy; computer-based instrumentation; computer science; software evaluation; educational technology systems/graduate education programs; computing in the USSR; use of computers with the learning disabled; study guides; database applications; desktop publishing; keyboarding; video graphics; adult and vocational education; business applications and industrial settings; LOGO; university computer education and microcomputer applications; educational computing research; university artificial intelligence programs; problem solving; staff development; computer songs; evaluating technology programs; computer learning month and computer clubs; laser disc applications; and pre-college instructional delivery systems. The text is supplemented by various figures and illustrations, and the full text papers contain references. An author index is provided. (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED304125

PROCEEDINGS

NECC '88

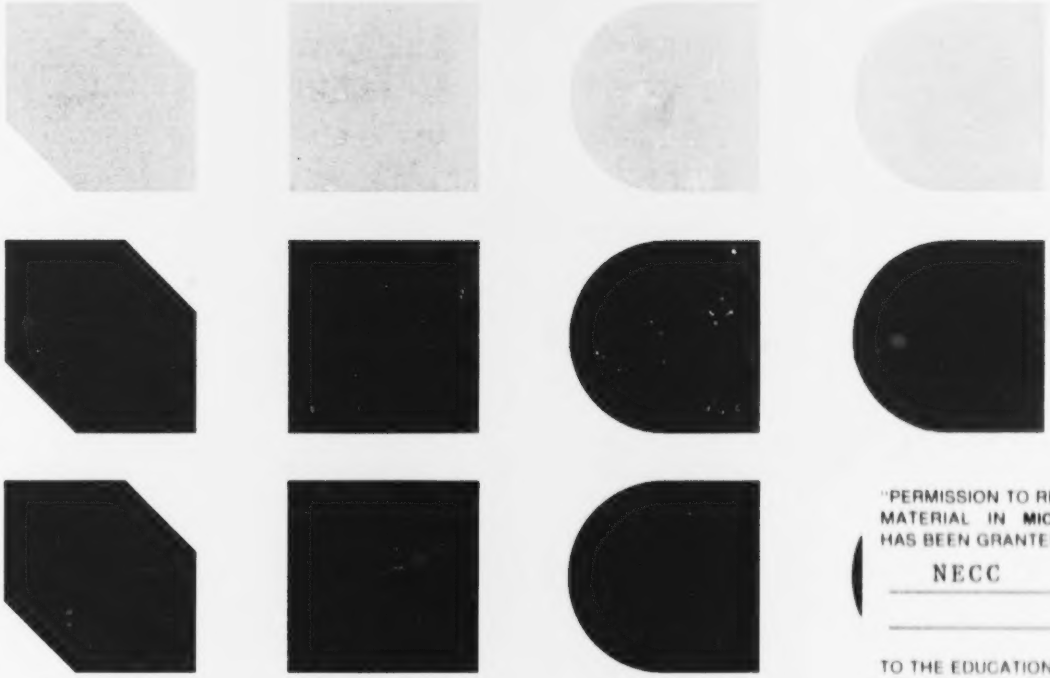
DALLAS, TEXAS JUNE 15-17, 1988

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.



"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

NECC

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

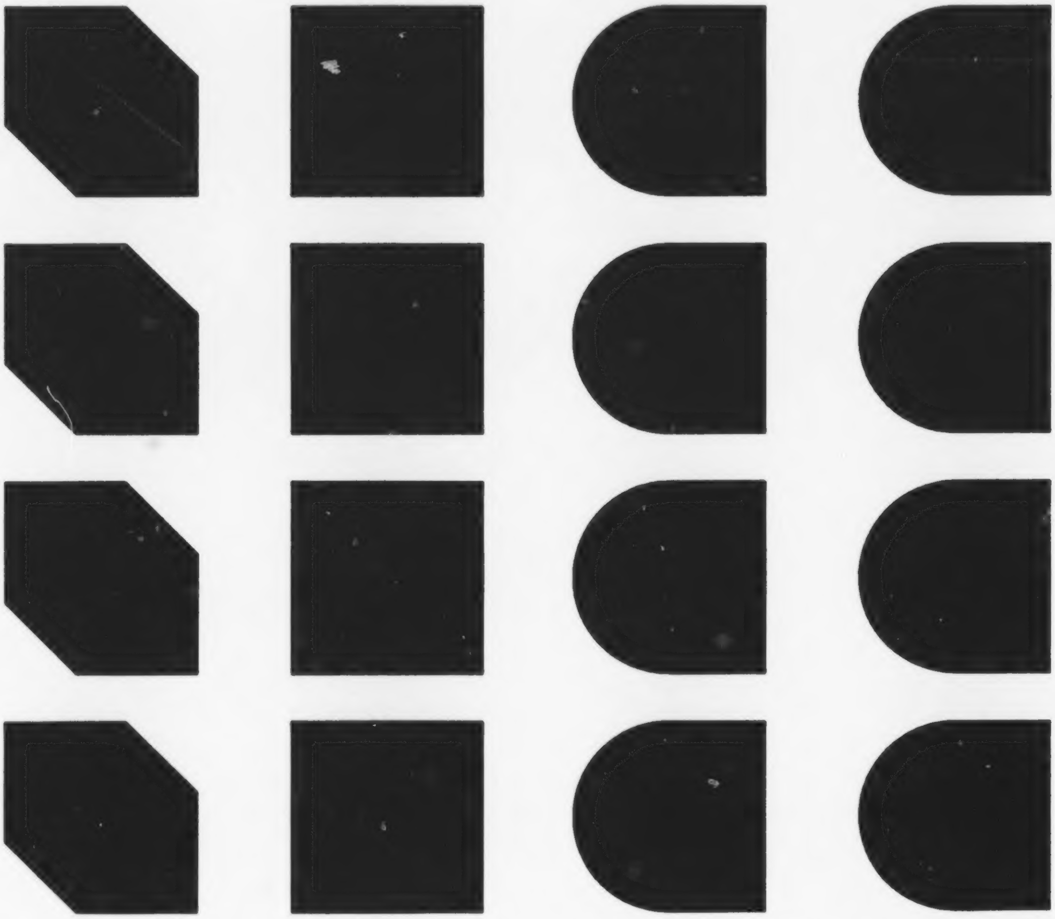
HOST: UNIVERSITY OF NORTH TEXAS

IR013698

PROCEEDINGS

NECC '88

DALLAS, TEXAS JUNE 15-17, 1988



HOST: UNIVERSITY OF NORTH TEXAS

William C. Ryan, Editor

PROCEEDINGS OF NECC '88

International Standard Book Number: 0-924667-45-1

Copyright © 1988: NECC

Published by:
International Council on Computers for Education
University of Oregon
1787 Agate Street
Eugene, OR 97403-1923
for
National Educational Computing Conference 1988
June 1988

NATIONAL EDUCATIONAL COMPUTING CONFERENCE

COOPERATING SOCIETIES

- American Educational Research Association (AERA)
- Association for Computers and the Humanities (ACH)
- Association for Computing Machinery/Special Interest Groups on:
  - Computer Science Education (SIGCSE)
  - Computer Uses in Education (SIGCUE)
  - University and College Computing Services (SIGUCCS)
- Association of Small Computer Users in Education (ASCUE)
- Educational Computing at Minority Institutions (ECMI)
- EDUCOM/EDUNET
- International Association for Computing in Education (IACE)
- International Council for Computers in Education (ICCE)
- IEEE Computer Society
- Society for Computer Simulation International (SCSI)

NECC STEERING COMMITTEE

Doris K. Lidtke
Chair
Towson State University

William C. Ryan
Vice-Chair
Swarthmore College

Ted Sjoerdsma
Treasurer
Washington and Lee University

Diana Harris
Secretary

Anita Best
ICCE
University of Oregon

Michael D'Amore
EDUCOM/EDUNET

Gerald L. Engel
Past Chair, Steering Committee
University of Connecticut at Stamford

Frank Friedman
Chair, NECC '87
Temple University

Nancy M. Ide
ACH
Vassar College

James Kerlin
ACM/SIGUCCS
Pennsylvania State University

John Lawson
ACM/SIGCUE
Towson State University

Jesse C. Lewis
ECMI
Norfolk State University

Michael Mulder
IEEE Computer Society
University of Portland

Nancy Roberts
SCSI
Lesley College

W. Waldo Roth
ASCUE
Taylor University

Thomas Saterfiel
AERA
Mississippi State Department of Education

Dennis Spuck
IACE
University of Houston

A. Joe Turner
ACM/SIGCSE
Clemson University

NECC '88 CONFERENCE COMMITTEE

Conference Chair
James L. Poirot
University of North Texas

Associate Conference Chair
Gerald A. Knezek
University of North Texas

Registrar/Treasurer
Patricia W. Warde
University of North Texas

Special Events
Denis A. Conrady
University of North Texas

Program Chair
Cathleen A. Norris
University of North Texas

Publicity Chair
James Adams
ACM/New York

Spotlight on Texas
Don Knezek
Texas Education Agency

Local Arrangements
Thomas C. Irby
University of North Texas

Associate Program Chair
Gerald L. Engel
University of Connecticut at Stamford

Special Sessions
Anita Best
ICCE

Workshops
Richard Austing
University of Maryland

Society Sessions
A. Joe Turner
Clemson University

Exhibits
Paul Katz
Nelda K. Owsiak
University of Oregon Continuation Center

Proceedings
William C. Ryan
Swarthmore College

AMERICAN EDUCATIONAL RESEARCH ASSOCIATION

The American Educational Research Association is concerned with the improvement of the educational process through the encouragement of scholarly inquiry related to education, the dissemination of research results, and their practical application.

AERA is the most prestigious and prominent international professional organization of approximately 14,000 educators; administrators; directors of research, testing, or evaluation in federal, state, and local agencies; counselors; evaluators; graduate students; and behavioral scientists actively concerned with educational research and its application. The broad range of disciplines represented by the membership includes: education, sociology, psychology, history, political science, anthropology, law, economics, philosophy, and statistics.

ASSOCIATION FOR COMPUTERS AND THE HUMANITIES

The Association for Computers and the Humanities (ACH) is an international organization devoted to encouraging the development and use of computing techniques in humanities research and education. Traditionally, ACH has fostered computer-aided research in literature and language, history, philosophy, anthropology, and related social sciences as well as computer use in the creation and study of art, music, and dance. As computing applications in the humanities have developed and broadened in the 1980s, the Association has expanded its scope to include areas from word processing to computer-assisted instruction in composition, language, history, philosophy, and anthropology, as well as computational linguistics and cognitive science, which overlap increasingly with work in the area of humanities computing.

Founded in 1977, ACH is the primary professional society for humanists who are involved or interested in any aspect of humanities computing. The Association provides a forum for continuing communication about humanities computing and strives to meet the needs of those who want to gain familiarity with both existing and potential applications of computers in humanities disciplines.

The heart of ACH is its quarterly newsletter, which covers the activities of the Association and its members and includes articles on various areas within humanities computing, news of projects and conferences of interest to ACH members, and reports on activities of governmental agencies and other organizations that affect computer-aided humanities research.

Computers and the Humanities, published by Paradigm Press, is a quarterly journal devoted to scholarship in the field of humanities computing. Subscription to *CHum* is included in the ACH membership fee.

ACH sponsors the bi-annual International Conference on Computers and the Humanities (ICCH), held in odd-numbered years, which brings together scholars from around the world to report on research activities and software and hardware developments in the field. Recently, ACH began to sponsor conferences and workshop on specialized topics in humanities computing, held in even-numbered years.

ACM SPECIAL INTEREST GROUP ON COMPUTER SCIENCE EDUCATION

SIGCSE became a special interest group of ACM in 1970. It currently consists of over 2000 members from the educational, industrial, and governmental communities interested in various aspects of computer science education. SIGCSE has goals of encouraging and assisting in the development of effective academic programs and courses in computer science and promoting research in computer science education.

The following are objectives of SIGCSE:

1. To provide a continuing forum for the discussion of common problems among educators and other computer scientists through organized meetings and symposia.
2. To publish a bulletin at least quarterly containing information aimed specifically at those interested in computer science education.
3. To work closely with the Education Board of ACM to insure implementation of effective education programs by the Association.

ACM SPECIAL INTEREST GROUP ON COMPUTER USES IN EDUCATION

The Special Interest Group on Computer Uses in Education (SIGCUE) of the Association for Computing Machinery (ACM) provides a forum for the discussion of ideas, methods, and policies related to all aspects of computers in the educational process. Established in 1969 its membership (over 1,500 persons) comes from many countries and numerous, diverse institutions.

SIGCUE publishes a newsletter titled the SIGCUE *Outlook*. SIGCUE also sponsors and organizes technical sessions at ACM annual meetings, the National Educational Computing Conference, and other national and regional meetings of interest to its members. A recent SIGCUE project consisted of a study group on national educational computer policy issues. This culminated in the publication by ACM of an edited volume entitled *National Educational Computer Policy Alternatives*. During the past year, SIGCUE cosponsored a project with the ACM Education Board and the International Council for Computers in Education a project that resulted in a model preservice course in educational computing for elementary education students. The background papers were published in the Summer '88 issue of *OUTLOOK*.

Among SIGCUE's goals are (1) publishing a current, substantive bulletin, (2) cooperating with other special interest groups or educational societies to promote attention to educational computing issues, and (3) helping to bring the technical expertise within ACM to bear upon educational computing generally.

ACM SPECIAL INTEREST GROUP ON COLLEGE & UNIVERSITY COMPUTING SERVICES

SIGUCCS provides a forum for those involved in providing computing services on a college or university campus. The topics addressed by SIGUCCS include managing campus computing, computing as it relates to the overall goals of the institution, and the state-of-the-art in various types of college and university computing services. Membership in SIGUCCS will help an individual to keep current with the world of collegiate computing services, and provides opportunities to discuss and share ideas and experiences with others.

Two annual conferences are regular activities of SIGUCCS. The Computer Center Management Symposium addresses the many aspects of managing computing on campus. This includes hardware, software, planning, finances, and personnel, to name a few. The User Services Conference deals more directly with the delivery of particular services to the higher education community. Tutorials on relevant issues are held at both conferences.

In other projects, SIGUCCS offers a Peer Review of the university computing function. Upon request of the computer center director, members of SIGUCCS will formally analyze and comment on different areas of your campus computing function. SIGUCCS also publishes a quarterly Newsletter. We consider the Newsletter our most important form of communication as it reaches all members and is subscribed to by numerous university computing centers. Conference proceedings are published either as separate documents or as part of the Newsletter itself.

THE ASSOCIATION OF SMALL COMPUTER USERS IN EDUCATION

The Association of Small Computer Users in Education (ASCUE) is a group of small colleges who work toward continued quality computer education and computer assistance in both administrative and academic areas. The purposes of the organization are: 1) to encourage appropriate uses of computing equipment and techniques for the improvement of its member institutions; 2) to supply its members with information on the most current computing methods; 3) to assist them in solving individual problems; 4) to cooperate with them in the utilization of the various small computers; 5) to cooperate with manufacturers, distributors, and suppliers in establishing and maintaining proper technical standards and in meeting new needs for special devices and systems.

EDUCATIONAL COMPUTING IN MINORITY INSTITUTIONS

ECMI is an organization run by a steering committee representing institutions whose student body reflects a large identifiable minority population. The objectives of ECMI are:

1. Computer literacy: To create among the faculty and administrators of the minority institutions an awareness and understanding of the strengths and weaknesses, uses and misuses, advantages and disadvantages, feasibility, practicability and limitations of computer applications in all aspects of society, including education.

2. Educational Computing: To narrow the gap which exists between the faculties in minority and non-minority institutions with respect to educational computing know-how and access.

3. Research Computing: To improve the computing facilities available to faculty of minority institutions for research purposes, particularly in those institutions offering graduate programs.

4. Technical assistance—consultants: To provide expert and impartial technical assistance to academic administrators of minority institutions on all phases of academic computing (instruction and research).

5. Education programs in the computer sciences: To improve the offerings of courses and degree programs in the computer sciences at minority institutions at all levels (e.g., introductory courses, minors, 2-year degree programs, 4-year degree programs, continuing education, graduate programs).

6. Computing facilities: To improve both quality and quantity of computing facilities available at minority institutions, because experience in the non-minority institutions has shown that an adequately staffed and equipped computer center for academic computing is essential to the success of previously stated objectives.

7. Direct student assistance: To increase the availability of minority staff for the computer centers and computer science education programs of minority institutions.

8. The need for a comprehensive program: To facilitate co-ordination and equitable distribution of funded activities to qualified institutions, associations, etc.

EDUCOM

EDUCOM is a nonprofit consortium of colleges, universities, and other institutions founded in 1964 to facilitate the introduction, use, and management of information technology. Through direct services and cooperative efforts, EDUCOM assists its members and provides leadership to the higher education community.

EDUCOM is funded by membership dues, service fees, and grants from foundations, corporations, and government. EDUCOM activities involve over 530 institutions and 90 corporations in the U.S. and abroad. In brief, they are: EDUCOM General Membership; the EDUCOM Consulting Group; EDUCOM Networking Activities including the Networking and Telecommunications Task Force (NTTF) and BITNIC, the BITNET Network Information Center; the EDUCOM Software Initiative; and the Corporate Associates Program.

For information on EDUCOM please write to P.O. Box 364, 777 Alexander Road, Princeton, New Jersey, 08540; 609 520-3340; or for general membership information via BITNET send to: MEMBER@EDUCOM.

INTERNATIONAL ASSOCIATION FOR COMPUTING IN EDUCATION

The International Association for Computing in Education (IACE), formerly the Association for Educational Data Systems (AEDS), continues to serve the educational computing field in three major areas: administration, instruction, and research. It is developing international ties with the ministries of education of six nations and also expanding its individual membership internationally. In addition, there is an increased emphasis on the administrative and instructional uses of microcomputers in higher education. Having weathered recent financial difficulties with the help of a grant from Epson America, Inc., the backlog of issues of the *Journal of Research on Computing in Education* has been produced, and the publication of the IACE Newsletter has resumed. In addition, IACE has entered into a management arrangement with the American Educational Research Association (AERA) to greatly reduce administrative costs. As a result of this arrangement, IACE will have its future annual conferences in conjunction with the AERA annual meetings, and the first IACE/AERA co-conference will be held April 5-9, 1988 in New Orleans. The theme of that conference will be Educational Computing: Practice, Policy, and Research. With the name change, new management arrangement with AERA, new membership structure, and new areas of emphasis, IACE will continue its 26-year tradition of serving the field of educational computing and looks forward to establishing cooperative efforts with other associations in the field.

THE INTERNATIONAL COUNCIL FOR COMPUTERS IN EDUCATION

The International Council for Computers in Education, ICCE, is a non-profit educational organization, with 12,500 Individual Members and 52 Organization and Associate Members.

ICCE publishes *The Computing Teacher*, C.A.L.L. Digest, and many booklets for persons interested in the instructional use of computers at the precollege level. These publications emphasize teaching about computers, teaching using computers, teacher education, and the impact of computers on curriculum.

ICCE has a substantial and growing professional outreach program. Four major components currently include:

1. Organization and Associate Members. Members publish newsletters and/or journals, hold conferences, and directly interact with their own members.

2. Professional Staff. ICCE has a professional staff who write, edit, participate in conferences, process orders, consult by phone or mail, etc. *The Computing Teacher* is put together by a full in-house production staff.

3. Ad Hoc Committees. Such a committee recently created the "ICCE Policy Statement on Software Copyright" and "Code of Ethical Conduct for Computer Using Teachers".

4. Special Interest Groups. ICCE has organized special interest groups for computer coordinators, teachers of educators, computer science educators, and videodisc-microcomputer users.

THE COMPUTER SOCIETY OF THE IEEE

The Computer Society is the world's largest association of computing professionals, with a total membership of approximately 90,000 computer scientists, computer engineers, and allied professionals. Society membership is open to IEEE members, associate members, and student members and to non-IEEE members who qualify for affiliate membership. An affiliate member is a person who has achieved status in his or her chosen field of specialization and whose interests focus in the computing field.

Every Computer Society member receives *COMPUTER*, a monthly magazine of general interest to computing professionals which also covers society news and events. Five specialized magazines and three journals are also available to society members as optional subscriptions and to nonmembers, libraries, and organizations. Magazines published by the Computer Society include *IEEE COMPUTER GRAPHICS & APPLICATIONS, IEEE MICRO, IEEE DESIGN & TEST, IEEE SOFTWARE*, and *IEEE EXPERT*. Research-oriented journals include *IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, and *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*. The Computer Society Press publishes nonperiodical literature, including tutorial texts and conference records. Last year 82 new titles were produced, and the society's catalog now contains almost 700 titles.

The society sponsors or cosponsors more than 100 conferences and meetings ranging from workshops and symposia with a few dozen participants to major conferences with many thousands of attendees. Over 30 technical committees offer the opportunity to interact with peers in technical specialty areas, receive newsletters, and conduct conferences and tutorials.

The Computer Society has over 100 local chapters throughout the world, and an additional 100-plus student chapters which provide the opportunity to interact with local colleagues and hear experts discuss technical issues. In addition, tutorials, educational activities, accreditation of computer science and engineering academic programs, the development of standards, and an international electronic mail network all play prominent roles in the society's activities.

THE SOCIETY FOR COMPUTER SIMULATION INTERNATIONAL

SCSI is the only technical society devoted primarily to the advancement of simulation and allied technology. It has a worldwide membership and a network of regional councils that covers the United States, Canada, and the United Kingdom.

The Society's original interest in the use of simulation in aerospace and engineering has expanded into many other industries, disciplines, and professions, including applications dealing with scientific, management, social, biological, and environmental problems. Artificial intelligence, robotics, and CAD/CAM are areas that have been specifically recognized as important to SCS members.

There are three major SCSI conferences each year. They are the SCSI Multiconference, the Summer Computer Simulation Conference (SCSC), and the Eastern Simulation Conference. There is also a new conference, the Simulation in Engineering Education Conference.

The National Educational Computing Conference thanks the following people for their contribution of knowledge, effort, and time as referees for the papers submitted for presentation.

Richard Adams
John Arch
Richard Austing
Bill Baird
Bruce H. Barnes
Nettie Bartel
George Bass
F. Samuel Bauer
James Beck
Henry Becker
John Beidler
Anita Best
Julie Bichteler
Rick Billstein
Ruthie Blankenbaker
William Blubaugh
Della Bonnette
James Bradley
Joyce Brennan
Sharon Burrowes
James Cameron
Doris Carey
Lowell Carmony
N. John Castellan, Jr.
Margaret Christensen
Frank Clark
James Clark
Richard Close
Ronald Cody
Betty Collis
James Comer
Leslie Conery
Frank Connolly
Sheila Cory
Bill Cotterman
George Culp
Bill Davis
Herbert Dershem
Jean Donham
J. Michael Dunlap
Larry Dunning
Kelly Eakin
Gerald Engel
Linda Ettinger
Ann Fidanque
Frederick Fink
LeRoy Finkel
Dexter Fletcher
David Flowers
Helen Frye
David Game
Robert Geist
Louis Gentry
Judith Gersting
Ronald Giedd
Allen Glenn
Kay Gore
Harold Grossman
Cindy Hanchey
Seymour Hanfling
Clifton Harris
Janet Hartman
David Hata
Sandra Hedetniemi
Henry Heikkinen
Heidi Imhof
Truman Jackson
Carolyn Jacobi
Larry Jehn
Dale Johnson
Norman Johnson
Tony Jongejan
Charles Kelemen
Henry Kepner, Jr.
Richard Kerns
Joyce Killian
C. Knox-Quinn
Barbara Kurshan
John Lawson
John Mark Leach
Laura Leventhal
Doris Lidtke
Joyce Little
Antonio Lopez
William Lyle
Harold Mac Dermot
Gary Marchionini
Stephen Marcus
Kathleen Martin
Judi Mathis-Johnson
Carla Mathison
Kathy Maury
Joerg Mayer
Donald McClain
John McGregor
Allen Melby
Sam Miller
Tony Mitchell
Andrew Molnar
Margaret Moore-Neiss
Michael Moshell
David Moursund
Chris Nevison
Bruce Norcross
William Park
Janet Parker
Jim Parry
Linda Petty
Philip Piele
Richard Plishka
James Poirot
Linda Polin
Gerald Pollard
Kathy Pon
Pat Powers
Gerard Rambally
Doris Ray
David Rine
Robert Riser
Nancy Roberts
Alan Roecks
Leroy Roquemore
Donald Ross
Waldo Roth
Greg Sales
Dean Sanders
Charlotte Scherer
Robert Seidman
Robert Shostak
Robert Slotnick
Pat Smith
Richard Smith
Dennis Spuck
Dennis Stevenson
Antonia Stone
Philip Stonesifer
Neal Strudler
Gail Sullivan
Paul Swan
Harriet Taylor
John Tenny
Robert Thomson
Sam Tumolo
Joe Turner
V. Lynn Tyler
John Underwood
Erwin Vernon
Henry Walker
Richard Wanderman
Jim Watson
John Wedman
David Weinman
Brent Wholeben
Billy Wood
Patti Zembrosky-Barkin

FOREWORD

This volume of proceedings of the Ninth National Educational Computing Conference (NECC '88) reports on innovations, trends, and research in the use of computers in a variety of educational settings. The conference provides an opportunity for the sharing of information through the presentation of reviewed papers, projects, panels, and special sessions. Selections for the final program were made from a large number of submitted proposals from authors with expertise in the computing field and commitment to advancing the effective use of computers at all educational levels.

The conference and these proceedings represent a tremendous commitment of time and energy from many persons who worked on planning and implementing this conference. I am appreciative of support extended by the following individuals:

- the National Educational Computing Conference Steering Committee members who provided guidance through years of conference planning , and especially the Executive Committee members, chaired by Doris Lidtke (Software Productivity Consortium), who contributed many additional hours of effort.
- the Program Committee of NECC '88 under the able direction of Cathleen Norris (University of North Texas, Denton) who spent long hours in the selection process which resulted in a first-rate program. Exceptional commitment to program excellence and effective teamwork was offered by:

  Gerald Engel (University of Connecticut at Stamford) who coordinated review and selection of papers; Anita Best (ICCE) who categorized and reviewed project presentations and special sessions, as well as organized post conference workshops;
  A. Joe Turner (Clemson) who demonstrated diligence and skill in soliciting outstanding society sessions;
  Richard Austing (University of Maryland) who provided an excellent array of preconference workshops; and
  Don Knezek (Texas Education Agency) who organized the exemplary "Spotlight on Texas" track.
- William Ryan (Swarthmore College) who completed the monumental task of organizing and editing the proceedings.
- Paul Katz and Nelda Owsiak (University of Oregon) who designed and managed the exhibits.
- Jim Adams (Association for Computing Machinery) who coordinated the publicity campaign for the conference.

Much of the real effort in planning a conference occurs at the local level. Special thanks are extended to the following individuals at the University of North Texas:

- Cathleen Norris, whose service far surpassed her Program Chairperson role.
- Gerald Knezek who served as Associate Chair and coordinated on-site activities.
- Patricia Warde who served as Conference Registrar and Treasurer.
- Paul Schlieve, Terry Holcomb, and Ray Coorpender who contributed service above and beyond the call of duty in organizing and providing A/V and computer services.
- Sharon Dumas and Dena Hilliard who provided outstanding and untiring secretarial support.
- The faculty, staff and students of the University of North Texas Department of Computer Education and Cognitive Systems who provided time, resources, and help in completing the large and small tasks necessary for the conference.
- The administration and faculty of the University of North Texas for their cooperation and work in hosting NECC'88.
- The family members of all the Conference staff who provided the love and encouragement we needed.

The Conference Committee finally would like to thank the attendees for their enthusiastic contribution to making NECC '88 a success.

James L. Poirot
Chair, NECC '88
University of North Texas
Denton, TX 76203

TABLE OF CONTENTS

2 W3-1 Logo and Mathematics (ICCE)
LOGO AND MATHEMATICS: ABSTRACT TO CONCRETE, T. Lough, I. Charischak, M. Rosen, N. Kovatch, J. L. McLaurin

3 W3-2 HyperMedia (ACH)
HYPERTEXT/HYPERMEDIA ON MICROS, G. Brett,D. Bantz, J. Bolter, B. Jones

4 W3-3 Secondary Computing Applications (Special)
COMPUTERS AND THE TEACHING OF LITERATURE, S. Marcus
THE HOST IN THE MACHINE: THE AI IN CAI, S. Marcus

6 W3-4 Computers in Social Studies (Special)
COMPUTERS AND THE SOCIAL STUDIES: PAST, PRESENT, FUTURE, M. Furlong, B. Hunter, S. Ambron

W3-5 Approaches to Teacher Education (Papers)
7 THE INTENSIVE MASTER'S PROGRAM AT TEACHERS COLLEGE: A MODEL FOR TRAINING TEACHERS TO USE COMPUTERS, H. Budin, N. Cunniff
13 TEACHING AND TECHNOLOGY: PREPARING TEACHER EDUCATORS TO INTEGRATE COMPUTER-RELATED SKILLS INTO PROFESSIONAL EDUCATION COURSES, L. Thomas

20 W3-6 Special Needs and Technology (Projects)
DATA BASE: A VEHICLE TO TEACH THINKING SKILLS TO STUDENTS WITH LEARNING DISABILITIES, C. Schmitt
A COMPUTER FOR EVERY STUDENT, D. Slater, B. Kosloski
FACULTY DEVELOPMENT WORKSHOP ON PERSONAL COMPUTERS, C. Wintz, W. M. Akalou

22 W3-7 Middle School (Projects)
PROJECT ACT (ACTIVITIES WITH COMPUTER TOOLS), J. Parker, S. Brown
ELECTRONIC SPREADSHEETS IN THE MIDDLE SCHOOL CURRICULUM, B. Raker
THE IDEA LAB, M. Rosen
DESIGNING A SCHOOL OF THE FUTURE—GUIDELINES FOR ADMINISTRATORS, R. Smith, A. Sassi

25 W3-8 AppleWorks (Special)
THE POWER OF APPLEWORKS, L. Hayward
APPLEWORKS GOES TO SCHOOL, J. Carlisle

27 W3-9 Professional Development (Special)
BEYOND INSERVICE TRAINING—TOWARDS PROFESSIONAL DEVELOPMENT, D. Watt, M. Watt, H. Olds

28 W3-10 Computer Science Curriculum: Secondary (Projects)
NETWORKING—A SHARING OF RESOURCES FOR COMPUTER SCIENCE AND BUSINESS EDUCATION, B. Sangston, D. Grantham, N. Sanchez
STUDENT PROGRAMMING OF REAL WORLD APPLICATIONS, W. Kertzman
THE ADVANCED PLACEMENT COMPUTER SCIENCE CURRICULUM PROJECT, L. Liden
DEVELOPING A METACOURSE FOR BASIC, S. Schwartz, D. Perkins, J. Kruidenier, D. Niguidula

31 W4-1 Legislative Action (ICCE)
INFLUENCING LEGISLATION: IMPLICATION OF THE OTA REPORT ON EDUCATIONAL TECHNOLOGY, A. Knight, L. Roberts, J. Stedman, R. Anderson, C. Blaschke, L. Finkel

32 W4-2 Elementary Math and Science (Projects)
IMPROVING SKILLS IN STORY PROBLEMS THROUGH WORD PROCESSING ACTIVITIES, V.

Morrison
MERGENCE OF SOFTWARE IN ARCHAEOLOGY WITH FOURTH GRADE STUDENTS, A. Ortiz
INTEGRATING THE COMPUTER INTO THE CURRICULUM AND OTHER SECRETS, D. Schmidt, L. Smith

34 W4-3 Telecommunications (ICCE)
AN EFFORT OF THE ICCE TELECOMMUNICATIONS INITIATIVES, C. Clark, L. Walters, D. Wydra, G. Hasloff

W4-4 College Level Computer Literacy (Papers)
35 ISSUES IN COMPUTER SCIENCE ETHICS: A SURVEY OF STUDENT VIEWS, L. Leventhal, D. Chilson
45 THE EFFECTS OF COMPUTER EXPERIENCE ON COMPUTER ANXIETY, S. Wallace
49 INFORMATION LITERACY: APPLICATION SOFTWARE AND CRITICAL THINKING—AN EXPERIENCE AND A PROPOSAL, D. Santomauro, H. Cinque

W4-5 Issues in Teacher Education (Papers)
56 CATALYSTS FOR CHANGE BASED ON TEACHERS' CONCERNS WITH MICROCOMPUTERS IN INSTRUCTION, R. Heller, C. Martin
62 TEACHERS AS COLLABORATIVE RESOURCES: PROFESSIONAL DEVELOPMENT THROUGH ASSESSING LOGO LEARNING, M. Watt, D. Watt
72 LOGO AND EQUITY: A NATURAL PARTNERSHIP, S. Yoder, R. Sutton, E. Fleming

75 W4-6 Secondary Content Areas (Projects)
THE TEXAS LEARNING TECHNOLOGY GROUP—A PHYSICAL SCIENCE INTERACTIVE VIDEODISC CURRICULUM PROJECT, C. Hardy, J. Young
THE DEVELOPMENT OF AN INSTRUMENT TO MEASURE THE IMPACT OF CALCULATORS AND COMPUTERS ON THE SECONDARY SCHOOL MATHEMATICS CURRICULUM, K. Maury
COMPOSITION APPROACH TO WORD PROCESSING, S. Merritt

77 W4-7 Special Needs and Equity: Middle-High School (Projects)
SEX EQUITY INITIATIVES IN MONTGOMERY COUNTY MARYLAND, D. Grantham, L. Odom, G. Tigani, B. Ryder
TWO PROJECTS USING TECHNOLOGY-ASSISTED INSTRUCTION TO TEACH ACADEMICALLY AT-RISK STUDENTS, R. Smith, E. Murphy
PROJECT TEMPT - TEACHING EXPLORATIONS WITH MIND-POWERED TECHNOLOGY, L. Ware
ESL FOR PRE-LITERATE MIDDLE SCHOOL STUDENTS, R. Price

80 W4-8 Statistics, Data Analysis, and Application Software (Special)
INTEGRATING TECHNOLOGY INTO THE TEACHING OF STATISTICS AND DATA ANALYSIS (GRADES 4-12), B. Bruce, S. Friel
PUTTING LONGEVITY INTO THE TEACHING OF APPLICATION SOFTWARE, T. Jongejan

W4-9 Computer-Based Education Policy (Papers)
82 RESEARCH AND EVALUATION TRENDS IN THE USES OF COMPUTER-BASED TOOLS FOR LEARNING AND TEACHING, B. Hunter
95 THE ROLE OF HYPERTEXT IN TRANSFORMING INFORMATION INTO KNOWLEDGE, C. Dede
103 INFLUENCE OF ROLE AND FUNCTION UPON STRATEGIC PLANNING FOR EDUCATIONAL COMPUTING POLICY INITIATIVES, B. Wholeben

110 W4-10 Secondary Data Processing and Computer Science (Projects)
COMPUTER SCIENCE CONTEST FOR HAWAII HIGH SCHOOL STUDENTS, E. Carey
DATA PROCESSING FOR THE VISUALLY IMPAIRED, J. Crummett, K. Matheson
TELETEACHING CLASSES IN CORRECTIONS EDUCATION, F. Ferrari, S. Keifer
TEACHING PROGRAMMING IN HIGH SCHOOL, S. Perelgut

113 Th1-1 Software Tools/AI Curriculum: College (Projects)
HYPED PSYCHOLOGY: USING MACINTOSH'S HYPERCARD TO TEACH JUNIOR COLLEGE

PSYCHOLOGY, C. Biediger
NEW TOOLS FOR STUDENT TRAINING AND RESOURCE MANAGEMENT IN THE COMPUTER LAB, J. Pope, P. Anderson, S. Stokes, P. Gordon
COMPUTER-BASED PRESENTATION GRAPHICS, N. Higgins
AN AI CURRICULUM FOR PRACTICING PROFESSIONALS, L. Roberts

115 Th1-2 Teacher Education (SIGCUE)
ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: PROJECT DESCRIPTION AND BACKGROUND PAPERS I, R. Taylor, K. Billings, E. Carlson, N. Cunniff, R. Spielvogel

118 Th1-3 Equity (SIGCUE)
PROJECTS TO INCREASE COMPUTER EQUITY, R. Anderson, A. Fredman, M. McMillan, E. Richman, J. Sanders

119 Th1-4 Pre-College Computer Applications (Special)
ASSOCIATIVE BROWSING: AN INTERDISCIPLINARY APPROACH TO TEACHING HIGHER ORDER THINKING SKILLS USING MICROCOMPUTERS AND VIDEODISCS, T. Kahn, A. McCormick
COMPUTERS, EDUCATION, AND CHILDREN'S LEARNING: THE REAL BASICS IN EDUCATION, T. O'Brien

Th1-5 Mathematically Based Issues (Papers)
120 DISCOVERING RELATIONSHIPS BETWEEN FUNCTIONS AND GRAPHS USING A GRAPHING PROGRAM, P. Cemen, J. Johnson
127 TEACHING BASIC AND PASCAL "LOGO" STYLE, S. Yoder

131 Th1-6 Telecommunications (Projects)
COMPUTER ASSISTED RETRIEVAL (C.A.R.) PROJECT: INSTRUCTIONAL USE OF DOW JONES, M. Bingham
DIALOG & EINSTEIN: EXTENDING THE MEDIA CENTER RESOURCES, M. Holloway
PROJECT DIAL: DEVELOPING INSTRUCTIONAL ACTIVITIES ONLINE, C. Denning
STATEWIDE E-MAIL: COMMUNICATING INSTRUCTIONAL ACTIVITIES, J. Campbell

132 Th1-7 Teacher Education (Projects)
GRADUATE AND UNDERGRADUATE CURRICULUM IN METHODS OF TEACHING COMPUTER SCIENCE, D. Carey
K-6 COMPUTER PROGRAMS—A SENSIBLE SOLUTION, C. Jacobi
WHAT'S HAPPENING IN COMPUTER SCIENCE TEACHER CERTIFICATION, M. Lintner
COMPUTER CONFERENCING A TEACHER ASSISTANCE MODEL, A. Colman, H. Hartle, S. Silverman

135 Th1-8 Computer-Based Instrumentation (Special)
SIMPLE AND INEXPENSIVE COMPUTER-BASED INSTRUMENTATION, B. Rouse

136 Th1-9 Computers in Mathematics (Special)
A YEAR-LONG NATIONWIDE FIELD EXPERIMENT ON THE EFFECTIVENESS OF COMPUTER USE IN GRADES 5 TO 8 MATHEMATICS CLASSES, H. Becker

137 Th1-10 Computer Studies: Secondary (Projects)
CHANGING THE FOCUS OF COMPUTER EDUCATION FOR THE COLLEGE BOUND STUDENT, C. Groninger
MAGIC NUMBERS GAME: AN ALTERNATE FIRST-DAY APPROACH TO A COMPUTER LITERACY CLASS, J. Taylor
A MODEL FOR THE TEACHING OF COMPUTER APPLICATIONS AT THE SECONDARY LEVEL, D. Hathaway

139 Th1-11 Software Evaluation (AERA)
THE CURRENT STATE OF THE ART IN THE EVALUATION OF EDUCATIONAL SOFTWARE: A VIEW FROM THE FEDERAL LABS, T. Saterfiel, D. Holtznagel, D. Bethke, M. Strezeski

140 Th2-1 Teacher Certification (ICCE)
TEACHER CERTIFICATION: A VIEW FROM THE STATES, T. Jongejan, D. Brittain, D. Knezek, B. Goodson

141 Th2-2 Computers in Education (SIGCUE)
ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: PROJECT DESCRIPTION AND BACKGROUND PAPERS II, J. Lawson, G. Bitter, R. Blankenbaker, B. Fedden, D. Watt

144 Th2-3 Teaching Thinking Skills (Special)
A MODEL FOR TEACHING THINKING SKILLS USING ONE COMPUTER, ONE CLASS, ONE TEACHER, AND PROBLEM-SOLVING SOFTWARE, R. van Deusen
DATABASES: TEACHING THINKING SKILLS IN CONTENT AREAS, J. Donham, M. Langhorne, D. Rehmke

146 Th2-4 Pre-College Applications of Technology (Special)
THE USE OF INTERACTIVE VIDEODISCS IN SCIENCE AND MATH INSTRUCTION, J. Woodward
THE ROLE OF COMPUTERS IN THE EVOLVING SCIENCE CLASSROOM, H. Imhof, R. Adams, R. McGrew

Th2-5 Issues in Writing and Social Science (Papers)
148 DESIGNING AN INTELLIGENT CURRICULUM PLANNER, K. McGraw, B. Brown
153 MAKING CONNECTIONS: LOGOWRITER, TELECOMMUNICATIONS, AND SOCIAL STUDIES, L. Foster

159 Th2-6 Pre-College Potpourri (Projects)
DATA BASES—LET'S GET RELATIONAL!, D. McClintock
TEACHER AND STUDENT ATTITUDES TOWARD COMPUTER ACCESS, J. Morgan
COMPUTERS AND CURRICULUM: DISTRICT-WIDE INVOLVEMENT OF TEACHERS, I. Varma

162 Th2-7 Educational Technology Systems/Graduate Education Programs (Projects)
SCHOOL-UNIVERSITY COLLABORATION: A FIELD-BASED DOCTORATE IN CURRICULUM AND INSTRUCTION WITH EMPHASIS ON COMPUTER INFORMATION SYSTEMS, R. Hinely, G. Ponder, B. Atkins
DEVELOPING A NEW GENERATION OF COMPUTER SPECIALISTS FOR SCHOOLS, P. Martorella
INSTRUCTIONAL COMPUTING: A GRADUATE EDUCATION CURRICULUM—5-YEAR EVALUATION, G. Tubb, F. Breit
CURRICULUM/INSTRUCTIONAL SYSTEM, J. Hart, D. Burrows

165 Th2-8 Speech Output (Special)
PRODUCING AND USING SPEECH-BASED SOFTWARE: A PANEL PRESENTATION, L. Anderson-Inman, B. Adler, M. Cron, M. Hillinger, R. Olson, B. Prohask, V. Gonzales-Baker

166 Th2-10 Tools for Teachers and Telecommunications (Projects)
DEVELOPMENT OF A STATE-WIDE CBE NETWORK IN MAINE USING NOVANET, O. Gaede
USING A CLASSROOM DATABASE PROGRAM TO PREPARE TEACHER-GENERATED TESTS, M. Land
TOOLS FOR TEACHERS, R. Sloan-Little

168 Th2-11 Computing in the USSR (SIGCUE)
COMPUTING IN THE USSR, K. Billings, F. Connally, C. Weiner, T. Lough

169 Th3-1 Teacher Education (ICCE)
THE REQUIRED COMPUTER EDUCATION COURSE: WHERE ARE WE AND WHERE ARE WE

HEADED? P. Moore, J. Friske, M. Lintner, B. Mathies, L. Thomas, J. Wiebe, C. Mlynarczyk

170 Th3-2 Computers in Education (SIGCUE)
ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: MODEL COURSE OUTLINE, S. Cory, H. Budin, D. Moursund, R. Taylor, J. Lawson

171 Th3-3 Study Guides/Learning Disabled (Special)
HYPERTEXT STUDY GUIDES: AN AID TO COMPREHENSION, R. Boone, K. Higgins
A MODEL FOR INTEGRATION OF COMPUTERS INTO THE CLASSROOM OF THE SEVERELY LEARNING DISABLED, C. McAvoy

173 Th3-4 University Potpourri (Projects)
GRADUATION AND ADVISEMENT SYSTEM (GRADS), J. Cook
A FOUR-YEAR RETROSPECTIVE ON REQUIRING MICROCOMPUTERS, P. Plourde
ESAU—EXPERT SYSTEM FOR ADVISING UNDERGRADUATES, W. Wehrs, T. D'Amico, D. Gering, I. Locketz, V. Schmitz
PUBLIC DOMAIN SOFTWARE: ITS IMPORTANCE IN TEACHER TRAINING; SOME SUBTLE POINTS TO PONDER, M. Batoff

Th3-5 Issues in Language Arts (Papers)

176 A HI-TECH LAB FOR TEACHING ENGLISH STUDENTS THE WRITING PROCESS, F. Ferrari, J. Knouse

181 Th3-6 Integration into K-12 Curriculum and District Implementation (Projects)
MARKETING SUPPORT SERVICES, R. Catalano, P. Lundberg
EVALUATING SCHOOL DISTRICT COMPUTER USE: AN ASSESSMENT PROCESS MODEL, C. Morton, D. Beverly
ALIGNING THE COMPUTER INTO THE CURRICULUM: MODEL AND IMPLEMENTATION, M. Warner, D. Johnson

183 Th3-7 Teacher Education (Projects)
TEACHER TO TEACHER TRAINING, ESTTI STYLE, B. Price, P. Kinnaman
PRE-SERVICE MICROCOMPUTER EDUCATION PROGRAM, J. Ray, M. Peyton
A COMPUTER-BASED SYSTEM FOR ADULT PRE-INSERVICE INSTRUCTION, S. Smaldino, T. Sherrill-Range
EDUCATIONAL COMPUTING APPLICATIONS AT BLOOMSBURG UNIVERSITY, D. Pratt

186 Th3-8 Database: Pre-College (Special)
DATABASE APPLICATIONS IN THE CURRICULUM, J.Poage, N. Strudler
FrEdBASE: A DATABASE PACKAGE FOR THE CLASSROOM, G. Butler

188 Th3-9 Educational Software/Telecommunications (Special)
EDUCATIONAL SOFTWARE FOR SCHOOLS: THE DUTCH CHOICE, J. Moonen
THE PLUTO NETWORK: AN EXPLORATION OF THE USE OF TELECOMMUNICATIONS IN TEACHER EDUCATION IN EUROPE, R. Gwyn, N. Longworth

190 Th3-10 Desktop Publishing (Projects)
SCHOOL DESKTOP PUBLISHING, J. Hillberry
DESKTOP PUBLISHING OF HIGH SCHOOL STUDENT PUBLICATIONS, S. Shuller
DESKTOP PUBLISHING OF A SCHOOL OR CLASSROOM MAGAZINE, C. Singletary, E. Cottingham
COMPUTERS HELPING INSTRUCTION AND LEARNING DEVELOPMENT (CHILD), S. Butzin

192 Th4-1 Reading and Art: K-12 (Projects)
HELPING RURAL ELEMENTARY TEACHERS TO INTEGRATE MICROCOMPUTERS IN K-8 READING AND LANGUAGE ARTS: A COURSE DESCRIPTION, M. French
THE COMPUTER AS AN ARTIST'S TOOL, K-12, L. Stumvoll
TECHNOLOGY SOFTWARE FOR READING LANGUAGE ARTS—AUTHORABLE COMPUTER ASSISTED INSTRUCTION IN READING COMPREHENSION SKILL, D. Tobin

ENCOUNTERING MUSICAL ISSUES AND MAKING THEM YOUR OWN, G. Greenberg

195 Th4-2 Computers in Education (SIGCUE)
ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: FUTURE DIRECTIONS, M. Watt, D. Moursund, J. Lawson, C. Berger, R. Taylor

196 Th4-3 Equity (ICCE)
ISSUES OF EQUITABLE ACCESS IN EDUCATION: THE INDUSTRY VIEWPOINT, A. Stone, A. Gonzales, W. Gaddis, M. Radabaugh, C. Edwards, J. Sanders

Th4-4 Administrative Matters (Papers)

197 TEA-NET: THE TEXAS EDUCATION AGENCY ELECTRONIC TELECOMMUNICATIONS NETWORK, P. Gehring, F. Noqueira
209 IS SOFTWARE LICENSING A SENSIBLE ALTERNATIVE FOR SCHOOLS?, R. Caldwell, L. Lockwood
215 MICROCOMPUTERS FOR TEACHER-PARENT COMMUNICATION: EXPLORING THEIR POTENTIAL EFFECTIVENESS FOR ELEMENTARY CLASSROOM TEACHERS, D. Bullock

Th4-5 Computer-Based Instruction (Papers)

224 THE EFFECTIVENESS OF MULTIPLE LINKED REPRESENTATIONS IN COMMON EDUCATIONAL SOFTWARE, D. Frye, E. Soloway
230 SOFTWARE TOOLS FOR A VIRTUAL CLASSROOM, E. Schreihofer, J. Foster, B. Gleason, H. Harting, S. Hiltz

237 Th4-6 Computer Science (Projects)
MICROPROCESSOR LABORATORY IN A SOFTWARE ORIENTED COMPUTER SCIENCE COLLEGE PROGRAM, A. Kornecki
"EMMEYLOU", AN INTEL 8085 TEACHING EMULATOR, AS USED IN COMPUTER ARCHITECTURE INSTRUCTION, E. Lewis, R. Baskerville
MARRYING THEORY AND PRACTICE: TEACHING FORMAL LANGUAGE CONCEPTS TO SMALL COLLEGE COMPUTER SCIENCE MAJORS, R. Rogers
A PEDAGOGY FOR PROLOG INSTRUCTION USING TURBO PROLOG GRAPHICS, G. Tubb

239 Th4-7 Keyboarding and Funding for Teacher Education: Elementary (Projects)
ELEMENTARY KEYBOARDING: FrEdTYPER AND OTHER ALTERNATIVES, G. Carnow
THE ACQUISITION OF MICROCOMPUTER KEYBOARDING SKILLS: DVORAK vs. QWERTY, M. Ford, V. Poe
KEYBOARDING: BEGINNING INSTRUCTION IN A PRIMARY GRADE COMPUTER LAB, R. Kerr
EDUCATION MEANS BUSINESS: PROJECT TEACH, H. Cowan

242 Th4-8 Video Graphics (Special)
VIDEO GRAPHICS: A DESCRIPTION OF THE WORKSHOP AND SOME CLASSROOM APPLICATIONS, B. Janssen, M. Marsh

243 Th4-10 Adult Education/Vocational Education (Projects)
A COMPUTER CURRICULUM FOR ADULT EDUCATION, M. Lintner
COMPUTERS IN A WORK PLACE LITERACY PROGRAM, D. Makings
VOCATIONAL TECHNICAL INFORMATION SYSTEM, V. Owens
SILICONE AND IVY—NEW PATHWAYS TO HIGHER EDUCATION, W. Seaton, J. Ice

245 Th4-11 Business/Industry (Special)
ARTIFICIAL INTELLIGENCE IN INDUSTRIAL SETTINGS, G. Yaverbaum, S. Dews, S. Breslawski, D. Hankins, R. Ziegler

246 F1-1 Potpourri (Projects)
ASSOCIATION OF HIGHER EDUCATION FOR NORTH TEXAS, PILOT PROJECT FOR A UNION CATALOG ON CD-ROM, K. Pearson Jagoe
PRODUCING INTERACTIVE VIDEODISC PROGRAMS ON A LIMITED BUDGET, R. Price
ARTIFICIAL INTELLIGENCE IN THE CLASSROOM, S. Kirkpatrick, B. Biglan

248 F1-2 Equity (ECMI)
EDUCATIONAL COMPUTING IN MINORITY INSTITUTIONS, W. DeLauder, S. DeLoach, G. Harrison, N. Watkins

249 F1-3 CAI and Equity: K-12 (Projects)
DEVELOPMENT OF CAI TUTORIALS TO ACCOMPANY THE ALABAMA BASIC COMPETENCY TESTING PROGRAM, J. Burney, M. Walker
SEX TYPING OF YOUNG CHILDREN'S BEHAVIOURS IN A COMPUTER ACTIVE PRESCHOOL CLASSROOM, A. Elliott
COMPUTER TECHNOLOGY IN AN EXTENDED TIME SUMMER SCHOOL PROGRAM, B. Woods
INTEGRATING THE ECIA CHAPTER 1 PROGRAM AT THE DISTRICT OF COLUMBIA PUBLIC SCHOOLS, A. Pitts

252 F1-4 Microcomputer Applications (Special)
ONE SCHOOL SYSTEM'S STRUGGLE TO ACHIEVE THE HIGHEST LEVEL OF COMPUTER INTEGRATION, J. Leonard, J. Harris, S. Bellamy

F1-5 Approaches in Computer Science (Papers)

253 ARTIFICIAL INTELLIGENCE PROGRAMMING: A STYLE FOR MEANINGFUL STUDENT ASSIGNMENTS, W. Adams
263 UNIVERSAL LOGIC IMPLEMENTER: A GENERAL PURPOSE TOOL FOR A DIGITAL LOGIC DESIGN LABORATORY, V. Malbasa, M. Manwaring, D. Seamans
273 THE SEARCH FOR A SUITABLE ACADEMIC MODEL FOR COOPERATIVE EDUCATION SUPERVISION IN COMPUTER SCIENCE, E. Adams

278 F1-6 University Computer Education (Projects)
AN INTEGRATED APPROACH TO UNIVERSITY FRESHMAN CURRICULUM, L. Hewitt, C. Bauer
A WRITING ASSIGNMENT FOR THE COMPUTER LITERACY COURSE, D. Olson
A DESIGN AND IMPLEMENTATION OF AN AUTOMATED TEST GENERATOR, M. Persson
COMPUTER-AIDED INSTRUCTION AT THE UNIVERSITY OF NEW HAMPSHIRE: PLANNING, IMPLEMENTING, EVALUATING, C. Kendall-Piel

281 F1-7 University Microcomputer Applications (Projects)
COMPUTER-BASED INSTRUCTION IN ENGLISH LINGUISTICS: A DEMONSTRATION AND DISCUSSION, J. Fisher
SYSTEM DOCUMENTATION TAILORED TO STUDENT NEEDS, R. Sabin
PREPARING A TEAM TAUGHT COURSE IN USER DOCUMENTATION, J. Talburt, W. Anderson
LEARNING DISABLED COLLEGE WRITERS IN A MICROCOMPUTER ENRICHED COMPOSITION CLASS: AN ANALYSIS OF PERFORMANCE OUTCOMES, N. Engen-Wedin

283 F1-8 Logo: Elementary (Projects)
MAKING MOVIES WITH *LOGOWRITER*, D. March
IMPROVEMENT OF CRITICAL THINKING SKILLS WITH LOGO AT THE SECOND GRADE LEVEL, P. Ohme, M. Knight
UNIVERSITY OF VIRGINIA'S *SURVEYS* AND *GRAPHWORKS* PROGRAMS TO BE USED WITH *LOGOWRITER*, M. Upton

285 F1-9 Educational Computing Research (Special)
RESEARCH IN THE APPLICATION OF COMPUTERS IN EDUCATION: TRENDS AND ISSUES, H. Becker, B. Collis, J. Moonen, D. Walker

287 F1-10 Telecommunications (Projects)
MAKING LOCAL AREA NETWORKS USEFUL INSTRUCTIONAL NETWORKS, K. Epstein
FREE-NET: AN OPEN ACCESS INFORMATION AND EDUCATION NETWORK, C. Myers, V. Carver
COMPUTER TECHNOLOGY AND THE DELIVERY OF NON-TRADITIONAL EDUCATION: CURRENT AND POTENTIAL APPLICATIONS OF THE OPTEL SYSTEM, K. Rodabaugh
ELECTRONIC PEN PALS: EDUCATION MAJORS CONVERSE WITH JUNIOR HIGH STUDENTS, C.

Scherer

289 F2-1 Business Applications (Projects)
USING FOURTH-GENERATION QUERY LANGUAGES IN THE MIS CLASS, H. Ligon
A DATABASE COURSE—INTEGRATING THEORY AND PRACTICE, G. McDonald, M. McDonald
INTEGRATED SOFTWARE: A COMPARISON OF THREE POPULAR PACKAGES, A. Schultz, M. Sutton, B. Zigalo, D. Neighbours

291 F2-2 University AI Program (ASCUE)
GETTING STARTED IN ARTIFICIAL INTELLIGENCE: A LIBERAL ARTS PERSPECTIVE, W. Roth

292 F2-3 Technology-Based Material (Special)
NEW METAPHORS FOR PRODUCING TECHNOLOGY BASED LEARNING MATERIALS, A. Bork

F2-4 Computer Science (Papers)
293 A COMPUTER-AIDED ADA COURSE, M. Tank, U. Pooch, B. Shirazi, J. Sampson
301 LPRINTER: AN EXPERT SYSTEM FOR PRINTER MALFUNCTIONS, J. Morrel
304 AN INTRODUCTORY COMPUTER SCIENCE COURSE USING LOGO: A CASE STUDY, E. Giangrande, P. Allmaker

F2-5 Problem Solving (Papers)
308 A THEORETICAL EXAMINATION OF COMPUTER PROGRAMMING AND MATHEMATICAL PROBLEM SOLVING, L. McCoy, N. Dodl
312 STUDENTS AS DEVELOPERS OF EDUCATIONAL COMPUTING PROJECTS: EXPERIENCES FROM CARNEGIE MELLON UNIVERSITY, C. Scheftic

F2-6 Research Projects (Projects)
IMPROVEMENT OF THE KNOWLEDGE BASE THROUGH USE OF A GRAPHICAL DATABASE, D. Brown
IMPLICATIONS OR RIGHT/LEFT BRAIN RESEARCH ON COMPUTER EDUCATION, V. Evans-Huse
THE RELATIONSHIP BETWEEN LEARNING PROGRAMMING AND PROBLEM-SOLVING ABILITY, J. Tu
HYPERTEXT AS A MEDIUM OF INSTRUCTION FOR COMPLEX MATERIAL, D. McGrath

F2-7 Problem Solving with Children (Papers)
325 POWER TOOLS FOR ALGEBRA PROBLEM SOLVING, N. Roberts, R. Carter, F. Davis, W. Feurzeig
332 INTELLIGENT TOOLS FOR MATHEMATICAL INQUIRY, R. Carter, W. Feurzeig, J. Richards, N. Roberts
337 GUIDED PROBLEM SOLVING, YOUNG CHILDREN, LOGO, AND IMPROVED HIGHER ORDER THINKING SKILLS, A. Shore

343 F2-8 Staff Development (Special)
EFFECTIVE STAFF DEVELOPMENT: PREPARING TEACHERS TO USE COMPUTER-AS-TOOL, D. Moursund, L. Connery, V. Johnson, B. Yates

344 F2-9 Music and Computers (Special)
MUSIC, MIDI, AND COMPUTERS, D. Latarski

346 F2-10 University Microcomputer Applications: Science/Health
A MODEL FOR TRAINING SCIENCE TEACHERS TO USE MICROCOMPUTERS, J. Ellis
COMPUTER ASSISTED INSTRUCTION IN HUMAN SEXUALITY, E. Randolfi, J. Lloyd
USING ARTIFICIAL INTELLIGENCE ENVIRONMENTS TO ENHANCE ELEMENTARY SCIENCE INSTRUCTION, M. Vitale, N. Romance

348 F3-1 Computer Songs (Special)
SONGS FROM THE AGE OF CLASSROOM COMPUTING, P. Rawitsch, D. Rawitsch

349 F3-2 Evaluating Technology Programs (Special)
EVALUATION OF LOCAL AND REGIONAL TECHNOLOGY PROGRAMS AND POLICIES, B. Collis,

S. Cory, G. Marshall

351 F3-3 Computer Learning Month/Computer Clubs (Special)
MAKING COMPUTER EDUCATION FUN THROUGH APPLE COMPUTER CLUBS, E. Kravitz
COMPUTER LEARNING MONTH 1988: WHAT IS PLANNED FOR THIS YEAR AND HOW CAN TEACHERS PARTICIPATE?, S. Bowman

353 F3-4 Integration: Elementary/Interactive Laser Disc Applications (Projects)
TECHNOLOGY AND THE FUTURE: A ELEMENTARY APPROACH, K. Becker, K. Hein, A. Taylor, M. Ellis
TECHNOLOGY AND THE FUTURE: AN INTERMEDIATE APPROACH, K. Becker, A. Taylor, K. Hein, M. Ellis
A COLLABORATIVE EFFORT USING QUEST TO DEVELOP INTERACTIVE MATHEMATICS PROGRAMS AND INTERACTIVE LASERDISC PROGRAMS FOR NON-NATIVE ENGLISH SPEAKERS, K. Bateman, H. Schwartz

354 F3-5 Potpourri (Projects)
COMPUTER LITERACY FOR ADMINISTRATORS, G. Prewitt
TEACHING SUPERPILOT WITH SUPERLESSON, D. Pratt

355 F3-6 Pre-College Potpourri (Projects)
USING FORMATIVE EVALUATION TO IMPROVE THE DESIGN OF INTERACTIVE VIDEODISCS, K. Hoelscher
TEMPLATES FOR LITERACY, A. Stone
IMPACT OF COMPUTER-BASED LEARNING ON ELEMENTARY STUDENTS' COGNITIVE ABILITIES: A UNIVERSITY-SCHOOL DISTRICT COLLABORATIVE RESEARCH PROJECT, K. Wiburg, M. Wilson

357 F3-7 University microcomputer applications: Health/Medicine (Projects)
ISWELL: AN EXPERT SYSTEM FOR WELLNESS ASSESSMENT AND EXERCISE PRESCRIPTION, J. Barlow, J. Morrell, C. Norton
THE DEVELOPMENT AND IMPLEMENTATION OF A MEDLINE DATABASE SEARCHING COURSE FOR 3RD YEAR MEDICAL STUDENTS USING PROGRAMMED INSTRUCTIONAL METHODS IN A LABORATORY SETTING, J. Sparks
USING COMPUTERIZED PATIENT MANAGEMENT PROBLEMS IN A MEDICAL SCHOOL CLASSROOM, J. Ullian, M. Noel
COMPARATIVE REVIEW OF HEALTH RISK APPRAISAL SOFTWARE, S. Brisk, E. Randolfi

360 F3-9 Pre-College Computer Applications (Special)
ADVENTURES IN MATHLAND: AN INTERDISCIPLINARY APPROACH TO K-6 MATHEMATICS INTEGRATING COMPUTERS, VIDEODISC, AND MANIPULATIVES, T. Kahn, M. Zauderer, A. McCormick
VARIATION IN AN INNOVATION: USAGE PATTERNS FOR THE VOYAGE OF THE MIMI, R. McCarthy

362 F3-10 Pre-College Instructional Technology Delivery Systems (Projects)
HAVING A BRAINSTORMING SESSION? TAKE A CAB (COMPUTER ASSISTED BRAINSTORMING), B. Ashby
CAN WE TALK? AN AUDIO RECORD AND PLAYBACK SYSTEM FOR THE APPLE IIgs FOR CREATING SPEECH-SUPPORTED LESSONS, R. Caldwell
SMALL AND LARGE GROUP APPLICATIONS OF INTERACTIVE VIDEODISC, L. Tuscher, P. Geyer
THE USE OF HYPERMEDIA AND HYPERCARD IN EDUCATION, M. Furlong, B. Franklin, J. Graeber, R. Knight, L. Schartz

SUPPLEMENT

W3-5

INFUSING THE COMPUTER INTO THE CURRICULUM: A WORKSHOP APPROACH, E. Vockell, H. Jancich

Th1-5

LEARNING MATHEMATICAL CONCEPTS THROUGH LOGO, S. Turner, M. Land

Th3-5

GRAPHICAL TECHNIQUES AS REVISION AIDS: AN EXPLORATION WITH LOGO, P. Freyd, K. Morris-Basik

F2-5

LEARNER CHARACTERISTICS AND LOGO PROGRAMMING: MATCHING STUDENTS WITH TEACHING METHOD FOR EFFECTIVE INSTRUCTION, S. MacGregor

PROCEEDINGS

OF THE

NATIONAL EDUCATIONAL COMPUTING CONFERENCE

JUNE 15-17, 1988

DALLAS, TEXAS

LOGO AND MATHEMATICS: ABSTRACT TO CONCRETE

Tom Lough, *Chair*
Founding Editor
Logo Exchange
Charlottesville, VA

ABSTRACT

In *Mindstorms*, Seymour Papert indicated that Logo began as an attempt to develop a Mathland for students. Because of this heritage and of a resulting mathematically rich structure, Logo has been used widely in mathematics teaching. Logo introduces students to many mathematical concepts at a wide variety of levels, ranging from number recognition and integer arithmetic through integral calculus and differential equations.

To date, the bulk of mathematics instruction with Logo has been accomplished using the somewhat abstract world of the computer screen. Recently, a product called LEGO TC Logo has been introduced which combines the abstract world of the Logo computer screen with the concrete world of LEGO building blocks. This makes it possible to use Logo to teach mathematics from both a concrete and an abstract perspective.

The panel is made up of educators who have used Logo both abstractly and concretely in their mathematics teaching. In their presentations, they will share their experiences, perspectives, concerns, and projections.

PANELISTS

Ihor Charischak
Council for Logo in Mathematics Education (CLIME)
Hastings-on-Hudson, NY

Nancy Kovatch
St. Anne's-Belfield School
Charlottesville, VA

Marian Rosen
Ladue Junior High School
St. Louis, MO

James L. McLaurin
EA Jones Elementary School
Sugarland, TX

SPONSOR: ICCE and SIGLogo

HYPERTEXT/HYPERMEDIA ON MICROCOMPUTERS

George Brett, *Chair*
The University of North Carolina Educational Computing Service
Research Triangle Park, NC

ABSTRACT

In this session panelists will discuss and demonstrate Hypertext and Hypermedia for instructional microcomputing in the humanities. Hypertext and Hypermedia are types of applications that permit students and faculty to have new ways of accessing any information that can be delivered by microcomputers and related peripheral equipment. This information includes data, graphics, live video, and sound stored on other devices such as CD ROM or laserdisks Because of the great variety of information modes in the humanities—text, graphic images, and sound—there is great potential for application of these concepts in the classroom.

The panelists will address the following issues: hypertext as a pre-writing tool; comparison of a hypertext application to a hypermedia application; and hypermedia that uses interactive laserdisk.

PANELISTS

David Bantz
Dartmouth College
Hanover, NH

Jay Bolter
University of North Carolina at Chapel Hill
Chapel Hill, NC

Robert Jones
University of Illinois
Urbana, IL

SPONSOR: ACH

COMPUTERS AND THE TEACHING OF LITERATURE

Stephen Marcus
Graduate School of Education
University of California
Santa Barbara, CA 93106

ABSTRACT

The computer-assisted teaching of literature involves a new set of terms, issues, implications, and applications. This presentation provides an overview of the state of the art and future directions for such developments.

Two related approaches, for example, are student-constructed databases and student-authored interactive fiction. In both cases, students can establish what critic Paul Hernadi refers to as "alternative itineraries...within the charted critical landscape."

Database software provides a specialized but flexible tool to help students think, talk, and write about literature. Properly used, it can (a) establish a collaborative learning environment, (b) demonstrate a powerful information-handling tool, (c) teach inquiry skills, and (d) put students in charge of the technology by allowing them to construct database structures that meet their own research needs.

Similarly, students can write (not just read) interactive fiction modeled on adventure games. Such software helps students develop their skills as information artists and knowledge architects, establishing a different sort of "database" as they study and write about literature. Traditional analytical perspectives like character, theme, point of view and audience analysis may be explored in unusual and inventive ways by allowing students to apply their insights to the production of literary landscapes. In doing so, students are accomplishing what Hernadi referred to as the necessity of "reintroduc(ing) the element of play into criticism."

Sherry Turkle at MIT has characterized the computer as "an evocative object, an object that fascinates, disturbs equanimity, and precipitates thought" (*The Second Self*). British author Daniel Chandler has noted that "the legacy of literature...has been a civil defense strategy by English teachers to protect children from the evils of modern industrial society...For such teachers the computer is often just another invasive influence in the classroom. It will no doubt be many years before there is any general shift in this stance" (*Computers and Literacy*).

We can certainly all stand to rethink our notions of how, and with what consequences, computers can affect the ways students think, talk, and write about literature. We should constantly reassess the problems and promise of technology. We should make explicit what we know and what we don't want to know about computers and the teaching of literature.

THE HOST IN THE MACHINE: THE AI IN CAI

Stephen Marcus
Graduate School of Education
University of California
Santa Barbara, CA 93106

ABSTRACT

Artifical intelligence (AI) can be considered from two perspectives with regard to computer-assisted instruction (CAI. The term as used here includes, but isn't limited to, drill and practice.) The first perspective has to do with AI that is *designed* into the software. The second has to do with AI that is *attributed* to software and computers in general.

For examples of attributed intelligence, we need only look back to Collette Daiute's early descriptions of how students using a word processor often felt there was an "audience" in the machine (*College*

Composition and Communication, 34(2) 1983). The word processor provided students with a peculiar kind of interactivity with their text; there was a "living" quality to the text (my word, not hers) that helped promote a kind of distance and audience awareness in the students. In a related regard, the culture at large still thinks of computers in terms of electronic "brains." Even when characterized as dumb ("they only do what you tell them to"), there is the implication that there's *some* sort of "intelligence" at work. No one, after all, speaks in the same terms about cars (i.e., no one bothers to say, "they only go where you tell them to go").

What I've been calling for some years "the host in the machine" is a feature not only of attributed intelligence but also of *designed* intelligence. An interesting example that bridges these categories is ELIZA (and its many variations), a psychological counseling program first developed by Joseph Weizenbaum at MIT (*Computer Power and Human Reason*). Even people who knew its nature, structure, and limitations sought ELIZA out because it was such a good "listener."

Of perhaps more immediate interest are programs designed for eduational applications. CODEKIT, developed in the United Kingdom by Daniel Chandler, is good at deciphering passages of text that you give it in coded form, and if asked it will tell you how it's going about its business. In this country, INTELLI-TYPE, from Electronic Arts, purports to be "so smart that it can monitor everything you type, evaluate your specific errors, and design customized lessons to help overcome them." My own efforts in the AI area resulted in AUTHOR! AUTHOR!, a game based on the old ANIMALS activity in which the computer tries to guess what's on your mind, and in which you can teach the program to become "smarter." Then there's Scholastic's A.I., in which (among other things) you can invent a game, its rules, and the criteria for winning. By merely playing the game with the computer, it will slowly come to understand the game, its rules - and most probably will start to beat you at your own game.

A sophisticated program like A.I. has a knowledge base, can acquire more knowledge, can look ahead to the implications of its actions, and can change its strategies to suit current situations. That's probably the easy part. What we really need are CAI-AI programs that adjust to students' cognitive and affective styles and to their reading levels.

We also need "hosts" that are considerate and have a sense of tact and perspective. In these regards, a lot of what's involved here has to do with the "voice" of the interface. English and language arts teachers have long been charged with helping students develop a repertiore of voices for their own writing (using the right voice is part of putting a good face on things-and their text is, after all, their interface with the reader). Teachers can now also turn their attention and expertise to helping publishers expand and refine their attempts to create really congenial AI, software that embodies intelligence informed with sensibilities.

I'm assuming that education involves interactions and that such interactions involve dialogue. Particularly when that dialogue is with computers, our vision should be of CAI-AI that not only allows but encourages students to hold up their end of the conversation.

COMPUTERS AND THE SOCIAL STUDIES: PAST, PRESENT, FUTURE

Mary Furlong, *Chair*
University of San Francisco
School of Education

ABSTRACT

What are 10 favorite "off-the-shelf" social studies programs? How are databases and other computer tools used in social studies today? What does the technology of computer-controlled interactive video offer social studies teachers in the future? What role will on-line communities have in the future of social studies?

PANELISTS

Beverly Hunter
Targeted Learning Corporation
Amissville, CA

Sue Ann Ambron
Apple Computer
Cupertino, CA

THE INTENSIVE MASTER'S PROGRAM AT TEACHERS COLLEGE: A MODEL FOR TRAINING TEACHERS TO USE COMPUTERS

Howard Budin and Nancy Cunniff
Teachers College, Columbia University
Institute for Learning Technologies
New York, New York

Introduction

The purpose of this paper is to outline the evolution and future directions of a program that can be an effective and exciting model for training teachers to use technology. The need for this kind of training is such that many alternative models can complement one another by supporting different needs of teachers. Training for practicing teachers is also crucial, and the model described here, a graduate M.A. degree program offered at Teachers College, Columbia University, provides a training structure which has proved to be popular.

Effective training of teachers requires that programs meet their special needs, which this degree program does in a variety of ways:

1. It provides necessary credentials for advancement in a teaching career.
2. It offers the support and services of a major university and teacher training college: the newest technologies, the finest research facilities, and instructors who are experts in the field become part of the educational process.
3. For students who are too far to travel regularly to New York, it provides a way to get a degree at Teachers College.
4. The program offers the excitement of immersion in a new environment and working with other teachers from around the country.
5. It allows half of the credits toward the degree to be earned via guided independent study at home, and it tries as much as possible to relate these credits to teachers' work situations.

Based on the interest in the program, the excitement and commitment it generates in its students, and the success its graduates have had, we present it as a model with strengths that should interest the entire teacher training community.

Program Background

In 1975 Teachers College, under the leadership of Professor Robert Taylor, began one of the nation's first graduate programs in Computing and Education. This 32 credit Master's degree program serves students who live in the New York area and those who are able to move to New York temporarily.

In response to inquiries from prospective students who were unable to come to New York for the academic year, we began the Intensive Master's Program in Computing and Education (IMP) in the summer of 1983, combining an intensive summer course at Teachers College with distance learning. Students in this program come to Teachers College for two successive July sessions of four weeks each, earning eight graduate credits each year. The remaining sixteen credits are earned through guided independent study before the first summer session begins and during the year between the two summer sessions. Thus in a period of just over one year students earn the M.A. degree in Computing in Education.

We limit the size of each IMP class to approximately twenty students due to considerations of space and intensity: we wanted a group to fit comfortably in a teaching lab and be small enough to get to know each other and the teaching faculty well. Because the actual contact time is somewhat brief, we felt it essential that students be in an environment where they would develop a strong working relationship with each other and with the faculty and staff during the first summer on campus. Limiting the size of the group facilitates interaction and helps ensure the building of communication links that sustain interaction during the independent study period.

The program was originally designed for students who already knew how to program and wanted to produce educational software. The course work was oriented towards instructional design, software evaluation and teaching computing. However, we continued to receive numerous inquiries about the program from teachers who knew little or nothing about using computers. Because of this continuing demand and because of our perception of the great need to train teachers who are new users of technology, in the fourth summer (1986) we redesigned the program, gearing it towards a more general degree in Computing in Education.

The IMP now accepts teachers of early childhood through high school as well school support staff such as librarians and media specialists, and gives them a thorough grounding in educational computing, including such topics as programming, software evaluation, and issues involving technology and society. The program requires no background in computing; in fact, we gear it towards the novice. The interest in this program has been such that we receive over 3000 requests for information about it each year.

Our goal has been to design and structure a program for teachers with whom we have direct contact for eight weeks, who do half of their course work in their homes around the country, and who have used computers very little if at all before they begin. We designed the program based on a set of pedagogical goals that are outlined in the next section. Through the first two years we have implemented a program based on these goals while expanding the scope and facilities of the program to serve our students better.

Key Factors in Training Teachers to Use Computers

How the teacher uses computer technology is the pivotal factor in the success or failure of educational computing (Amarel, 1983). What the teacher knows how to do *confidently* with computers, s/he will do. But computers are unlike other educational tools, and good educational use of computer tools can require new skills and even new ways of thinking for teachers. Therefore teachers must be trained specifically and thoroughly to take command of the technological tools they need if they are to use computers to their potential (Sutphin, 1987).

We perceived our Intensive Masters Program as a vehicle for modeling the kind of teacher training that can foster successful use of computers in schools. In this section we outline a set of key factors in training teachers in general and specifically in the use of technology. These describe ideal characteristics of teacher training. In the following section we show how the implementation of the IMP so far and our plans for its evolution begin to exemplify them.

1. Active teacher involvement and learning to build competency

Recent studies have identified, as among the most important characteristics of successful teacher training, that teachers get a sense of their *own efficacy* through active use and exploration of materials, and that, in the case of technology, teachers become *active users*, learn *from each other*, and that learning should extend over a *long period of time* (Corwin, 1983; Sturdivant, 1986; Utrecht, 1983). These factor is vital in any kind of teacher training and, we contend, even more so in training to use technological tools that are unfamiliar to teachers. In this case the training can involve extensive and intensive practice in learning how the tools work *before* considering their best uses in curriculum. The goal is to have teachers become competent users of computing and, thus, confident that they can make decisions concerning best uses in their own environments.

2. Using technology in developing model curriculum

Educators have repeatedly pointed out that computer technology can facilitate the attainment of many curricular goals, such as:

-*Access to information* can be dramatically increased through use of databases (Hunter, 1987).

-*Thinking and problem solving skills* can be promoted through simulations and use of tool software (Lengel, 1987; special issue of **Computers in the Schools**, 1986).

-*Global understanding* can be enhanced through electronic access to international data (Kendall and Budin, 1987).

-*Computer networking* can link students to other students around the world as well as provide access to information (Rhodes, 1986).

-Computers can support the development of a curriculum that is meaningful for an information-based society by *connecting domains of learning* (Cleveland, 1986).

-Computers can provide a unique opportunity for setting up *social and interactive* learning situations, in which students learn through collaboration with their peers and take command of their own learning (Budin, Taylor, and Kendall, 1987).

These are a few examples of the ways in which computer technology, in supporting curricular goals, may also change traditional school learning structures. Separate curricular areas, for example, may become seen as more closely interconnected, and teachers may begin to perceive their roles more as facilitators of learning than as sole imparters of information (Taylor, 1987). There is evidence that many teachers using computer labs for the first time have recognized such a difference in their roles, but have not been specifically prepared for the consequences of such a change (Budin, H, J.A Eisler, F. Guerrero, and J. Schoener, 1987).

Thus, training teachers to use educational software is only one step in successfully integrating computers into curriculum. Further steps involve a thorough grounding in using tools like simulations and databases and networking over an extended period of time, a continuing exploration of such tools with students, and a careful consideration (informed by practice) of whether and how to use these tools in particular educational settings.

3. Long-term collaboration

The potential for failure in brief training programs is high (Corwin, 1983). As implied in the first two factors discussed above, successful training should involve teachers over an extended period of time. Moreover, a successful program should provide continued support *even after the training is through*. Ideally, it should also collaborate with school and district leadership as well as with individual teachers to ensure support throughout the system.

4. Networking

> "Most teachers and administrators...work in relative isolation from their peers. As we discover how effective schools and districts function, we see the need for peers to have access to each other's experience as well as to their authoritative practical knowledge. And we need to be able to tap these networks as problems occur. Here is a major area for technology: providing tools for connecting educators with

each other...." (Rhodes, 1986)

Networking in educational settings takes many forms, all of which are be important vehicles for teacher support. Traditionally, the most common school-based networks are one-to-one and small group communication within a grade or discipline of a single school and interschool structures of communication through organizations and meetings. More recently, electronic networking has become possible, linking teachers and students within a school, across a city, state or even across the country. Electronic networking can instantly connect people at great distances, enable exchange of information and opinion through mail and conferencing, and allow quick feedback and help for problems encountered. It is one of the most important contributions that computer technology makes to the facilitation of communication between and among educators. Computer networking can, and already has begun to, connect teachers around the nation and effectively reduce some of the sense of working in isolation.

Supporting Teacher Training Through the IMP

In this section we describe several key features of the IMP and discuss how each relates and contributes to the teacher training process. In keeping with out goal of describing this program as a model for training, we discuss the program in terms of its main features.

1. *Course structure*. Students spend two four-week summer sessions at Teachers College. Although course details may vary slightly from summer to summer, we present a core curriculum in the first summer:

-**Programming I**: an introduction to programming concepts, structures and problem solving, using FPL (First Programming Language, developed here at Teachers College by Professor Taylor) and Pascal;

-**New Technologies for Learning**: an introduction to a variety of audio, visual, optical storage, and communication technologies focusing on current and potential uses in education;

-**Software Evaluation**: combines an examination of educational software with an examination of learning theories and their pedagogical applications, so that students can begin to design computer-using curriculum appropriate to their specific educational settings.

A course surveying the history of communications and computing in the Western world, or one that considers the impact of new technologies on society and education.

The second summer contains various options, depending on students' areas of teaching and interests. All students continue the study of programming in Programming II. Students may elect to take a course in LOGO, The Computer in the Elementary School Curriculum, or other courses involving computers and writing, computer graphics, or computers in administration.

All of these courses, even the most theoretical, involve the use of computers. The programming courses are taught in an integrated lecture-lab situation, use a computer teaching lab almost all of the time. Others, such as Software Evaluation, combine demonstration and discussion with occasional in-class lab time, and require considerable use of computers for assignments. All courses involve students in the development of projects, usually collaborative ones, using computer tools. During the first few days of the program, we conduct intensive workshops for the first year students, teaching them to use the tools they will need - a word processor, a database manager, and DOS - so that from the beginning they can use these tools in class projects and assignments.

The first summer's courses contribute to those aspects of teacher training identified as the first two key factors in the previous section. Our primary goal in the intensive summer session is to build our students' confidence in themselves as competent computer users - they should develop a mastery over the tools they will need to use in their own school work as well as in assignments for the IMP during the year. Further, through their courses the students learn about state-of-the-art technological applications, such as video and audio interfaces, and practice using them. At the same time, all of the courses address the question of how technology is being used, and could be used, in school curriculum. Thus while learning to use the technology our students are also reflecting on what its best uses might be.

2. *Collaboration between students*. Another aspect of active involvement is the interaction our students have with each other during the program. Instructors in the IMP promote and encourage such interaction through group projects and assignments. Since each group consists of approximately 20 students and almost all of them live in the same campus dormitory, group members quickly become quite close, through classes, organized after-hours discussions, and extracurricular events like explorations of New York or trips to the beach on weekends.

As part of their course work, the second-year students interact with the first-year students by conducting seminars and workshops. This has been a particularly successful aspect of the program for several reasons. The students conducting the workshops do so in conjunction with a strand of their course-work focusing on *Teaching Computing*; this assignment provides them with an opportunity to design materials, test ideas and interact with beginners. The seminars/workshops provides yet another opportunity for both groups of students to work collaboratively.

The "intensive" aspect of the program makes it particularly important that the students begin to work together immediately for two primary reasons: 1) there are many assignments, and 2) they must be completed in a relatively short time period. We require collaboration for many projects and encourage it for most others, with the idea that once the students begin to collaborate, they will continue to do so during the independent study year and even after graduation. We are beginning to see the realization of this in the types of independent study projects that students are designing. Many students propose joint projects even though they live and work in different parts of the country. They communicate using a variety of traditional means as well as via electronic mail and exchange of disks. Thus the student interaction and collaboration fostered in the first summer session bears fruit in terms of long-term collaboration and networking between present students and alumni of the program. It may even serve as a model for how teachers could work and learn together in the students' own school settings.

An especially interesting aspect of the program is the geographic diversity of the students. This past summer we had teachers from Alaska, Puerto Rico, Georgia, Texas, North Dakota, New Hampshire, and other parts of the United States. Learning from and with people of diverse backgrounds can teach a lot about differences as well as similarities between people and regions. We find that the experience of getting to know such a diverse group of people is exciting and often creates strong bonds between students.

3. *Independent study.* The purpose of the independent study is to provide continuity between the summers and to connect the learning experiences of the first summer with real teaching situations to the extent possible. All independent study projects are guided by the Teachers College faculty and staff and require intermittent contact during the semester. In one strand of the independent study students review and extend their programming knowledge. In another, they continue/expand their software evaluation work, focusing on using selected software in curriculum. In another, they are guided while they design individual project(s) for applying computing to their own situations. This last may involve a variety of options: creating a computer program for their own class, school or district; observing students using computers and analyzing the process and outcomes; studying one aspect of computer use, such as simulations for social studies; or researching a type of software with an eye toward making decisions or recommendations for their school. The point is to connect the summer learning to an actual school situation in whatever ways students can do this.

We continually attempt to connect at least some of the independent study assignments with the course structure of the second summer session. Thus the training can be viewed as a continuous 14 month period. During the school year we make every effort to stay in touch with students, because we know that the experience of working alone on six graduate credits per term while also working at a full time job can be much more difficult than learning with a group of people in an intense and exciting campus situation. We have three main methods of keeping in touch during the year:

-The course and independent study instructors write to students periodically, and students telephone whenever they want. This provides feedback on problems and assignments as well as some real contact for the students.

-In the middle of the school year we have a weekend meeting in New York for anyone who is able to attend, with a combination of informal discussions, formal problem solving sessions, guided tutorials, and socializing.

-We have instituted our own electronic bulletin board system, which we discuss separately below.

Ensuring successful distance learning is a difficult task; after the intensive summer session our students often find themselves back at extremely demanding jobs, with families to think about, and without the personal support and excitement of the group process. We find that we can use the success of the summer session to build on three of our key factors. First, the independent study should continue the active involvement with the technology that the summer began: some assignments require students to use computer tools such as word processors and databases, and others encourage students to use technology actively in their schools. In these ways, the competencies built during the summer should be extended. Second, assignments encourage students to use, or consider the use of, technology in their curricular practices, through evaluating software, designing curriculum projects, or doing research on what other schools are doing with computers. Third, the support we offer during this period is long-distance, increasingly through our computer network (described in the next section): we respond to students' questions, we offer informational bulletins, and students and staff all correspond with each other.

4. *Networking*. Last year we realized that the support mechanisms we had provided were too limited. Students with problems in programming, for instance, often found it difficult to discuss bugs or misconceptions over the telephone; and communication *via* through the postal service's *snail mail* can be very time-consuming. Additionally, we emphasized group problem solving during the on-campus sessions and the students indicated that more than anything, they missed that interaction during the interim year, when working at their often remotely located home towns. So, we wanted to devise a way for students to communicate as easily with each other as well as with us, while at the same time being cognizant of the cost incurred by telephoning frequently over long distances.

Because we believe that the building, support and use of both human and electronic networks *is* a key factor in the success of a training program, we installed a bulletin board system, using RBBS (Remote Bulletin Board System) software on an IBM AT. We have this running as a 24-hour a day service, presently on a single line. We are in the process of obtaining an 800 (free call-in) telephone number so students will not have to pay for the connect time, thus further encouraging its use.

At present, nearly all of the group of students who began the program last summer have registered on the bulletin board. We consider this an accomplishment since most of them had to buy their first computer and modem and learn about communication software upon their return home from Teachers College. We are presently using the system for simple message exchange and public posting of bulletins, but we are in the process of setting up special interest group mechanisms soon, and plan to explore the possibility of expanding to a multi-line system so that we can conduct on-line conferences.

5. *Special interest groups*. Because of the varied interests among the groups here last summer, we have begun to build a special interest group structure into the IMP. Each SIG has one or two students coordinating it along with an instructor in the program. At present we have three SIGs: one for computer coordinators, one for those interested in using databases in education, and one for those interested in teaching Logo. These groups will communicate regularly through newsletters put together here at Teachers College, and via the bulletin board. The SIGs encourage both networking and long-term collaboration between present and former students in the program.

The computer coordinator SIG deserves a few words of explanation. The IMP, as we have noted, accepts students who have little or no computer background. We have found, through surveying incoming students, that few of them plan to leave their classrooms and take on administrative or coordinating responsibilities as the result of this program. Nevertheless, a survey done by one of the 1986 group as he and his classmates were graduating showed that over half of the group had already changed jobs (either within their own school or district or to another district) to one involving more responsibility for computer use. Several had relocated across the country to become computer coordinators. The majority of students in the program are involved in the computer coordinator SIG.

6. *IMP Alumni*. Several program aspects discussed above, especially the SIGs, imply involvement that continues beyond the 14 months of the formal degree program. By the end of the second summer, most of the students have formed close personal and professional ties and they do not want this involvement to end. Therefore, in the summer of 1988, we plan to hold the first annual two- or three-day meeting of all IMP alumni. We want all of our graduates to stay involved with us, with each other, and with the special interest groups. We have a feeling that the kinds of bonds created here, coupled with the kinds of computer and curriculum work our graduates are doing across the country, can lend strength to the ongoing program and to all the students in it. More importantly, as our alumni assume roles of leadership in schools around the country and the world, continued involvement in a professional growth and education program will help ensure that a network of computer educators is established.

Future Program Goals

Building on what the IMP accomplished so far, there are several directions for future development:

1. **Networking**. We aim to expand the size and scope of our computer network, to include such features as independent study conferences and electronic transfer of students' assignments. As teachers become more comfortable using computer networks, we also plan to have them develop curriculum for their students' use of networks.

2. **Collaboration** with school districts. Already begun on a limited basis, we want to encourage school districts to work with us in several ways: by sending teachers to the program and supporting them financially; by granting sabbaticals during the school year so teachers can devote a considerable amount of time to work on the program; and by having teachers work with district leadership during their independent study so that their work can be relevant and useful to the district.

3. **Long-term work with students**. As our graduates begin to move into positions of responsibility for the use of technology in their schools, and begin to design and implement programs for such use, we want to work with them in several possible ways: assisting in program design; conducting joint research projects on development and implementation of programs; and on disseminating information about what they are doing.

4. Expansion and development of the SIGs. In the future there will be more SIGs as part of the program, and more members in each. As the number of activities grows, there will need to be better systems of communication within each SIG and between SIGs and the rest of the program. This will include use of the computer network as well as printed newsletters for each group. Another objective is to begin coordinating SIG activities with similar groups of other organizations nationwide.

5. **Research opportunities**. Our contacts in schools across the country give us a window on how schools with wide range of socio-economic and cultural diversity use technology. This provides a rare opportunity for conducting research on the implementation of computing programs and on the educational effects of computing.

The Intensive Master's Program at Teachers College is one unique effort to apply key principles of teacher education to the training of teachers for using technology in education. Our experience suggests that this model is successful and popular. We have already trained a small but influential cadre of teachers across the U.S. and several foreign countries. We use technology to teach about technology and thus support the adage that "teachers teach as they were taught."

References

Amarel, M. (1983). The Classroom: An Instructional Setting for Teachers, Students, and the Computer. In Wilkinson, A.C. (Ed.), **Classroom Computers and Cognitive Science**. New York: Academic Press.

Becker, H.J. (1986-87). Instructional Uses of School Computers: Reports from the 1985 National Survey. A series from The Johns Hopkins University, Center for Social Organization of Schools.

Boyer, E.L. (1983). **High School: A Report on Secondary Education in America**. New York: Harper and Row.

Budin, H., R.P. Taylor, and D.S. Kendall. (1987). Computers and Social Studies: Trends and Directions. **The Social Studies**, 78 (1), 7-12.

Budin, H., J.A. Eisler, F. Guerrero, and J. Schoener. (1987). A Report on Phase One of New York City's MAC/VIP Computer Labs. A paper delivered at the American Educational Research Association's Annual Meeting, April 1987.

Cleveland, H. (1986). Educating Citizens and Leaders for an Information-Based Society. **Educational Leadership**, 43 (6), 62-64.

Computers in the Schools, 3, (1). Special issue on **The Computer as an Educational Tool**, Spring 1986.

Corwin, R.B. (1983). Looking for a Model of Computer Literacy Training. **Journal of Staff Development**, 4 (2), 6-13.

Goodlad, J.I. (1984). **A Place Called School**. New York: McGraw-Hill Book Company.

Hunter, B. (1987). Knowledge-Creative Learning with Data Bases. **Social Education**, 51 (1), 38-43.

Kendall, D.S. and H. Budin. (1987). Data at Your Fingertips: The Computer and a Global Perspective. **Access** no. 67, a publication of Global Perspectives in Education, December/January 1987, 7-10.

Lengel, J.G. (1987). Thinking Skills, Social Studies, and Computers. **The Social Studies**, 78 (1), 13-16.

Rhodes, L.A. (1986). Technology: Reform's Missing Tools. **Educational Leadership**, 44 (1), 87.

Sturdivant, P. (1986). Planning and Training for a New Educational Delivery System. **Educational Leadership**, 43 (6), 38-40.

Sutphin, D. (1987). Educating Teachers in Instructional Applications of Microcomputers. **T.H.E. Journal**, 14 (6), 54-58.

Taylor, R.P. (1987). Computers for Intellectual Regeneration. **Social Education**, 51 (1), 34-36.

Tucker, M. (1985). Computers in the School: What Revolution? **Journal of Communication**, 35 (4), 12-23.

Utrecht, G. (1983). A Staff-Developed Plan for Computer Staff Development: Teachers Teach Each Other. **Journal of Staff Development**, 4 (2), 29-34.

TEACHING AND TECHNOLOGY: PREPARING TEACHER EDUCATORS TO INTEGRATE COMPUTER-RELATED SKILLS INTO PROFESSIONAL EDUCATION COURSES

Lajeane G. Thomas
Louisiana Tech University
Ruston, Louisiana 71272

ABSTRACT

Current literature describing the use of computers in educational settings supports the premise that to effectively use computer-based technologies in instruction, teachers should complete computer education coursework preparing them (1) to teach topics typically encompassed in computer literacy classes and (2) to use microcomputers to enhance the delivery of the K-12 curriculum in math, science, social science, language arts, etc. On the forefront of legislation to support this contention is California State Assembly Bill 1681, which specifies that teacher candidates seeking a clear teaching credential must satisfactorily complete computer education coursework that includes "general and specialized skills in the use of computers in educational settings." (Blurton, 1986)

Often higher education faculty who train teachers may view computer-based requirements separate from the existing methods coursework. The California Computer Education Advisory Panel, however, suggests that computer methods be incorporated into the preservice methods coursework as an integral part of the instructional process for each subject and encourages the development of separate computer-based technology education classes. (Blurton, 1986)

The difficulties for Colleges of Education (COE) in Louisiana and nationwide in implementing such a solution are: (1) Many COE faculty have not yet developed the technological skills to feel comfortable with incorporating computer activities into their methods courses; (2) They are not yet aware of strategies, methods, skills, and activities that particularly enhance instruction in individual subject areas; and (3) Their awareness of software applicable to the subject areas is limited. Technology training for teacher educators is clearly needed.

This paper will discuss the planning, implementation and results of a teacher educator technology training program implemented in the College of Education at Louisiana Tech University and identify affective, cognitive, and behavioral factors found to affect computer use. Teacher educator and preservice teacher competencies are suggested and statistical analysis of pre and post evaluation or students are reported.

INTRODUCTION

The teacher training effort in computer education is one of great magnitude and one that must be addressed immediately. The Association for Computing Machinery (ACM) Elementary and Secondary Schools Subcommittee (1981) recommended efforts in teacher training be aimed at the areas:

1. Computer training of preservice teachers;
2. Computer training of practicing teachers;
3. Development of graduate programs on computers in education; and
4. Research on computers in education.

The California Computer Education Advisory Panel and other noted computer educators (Rogers, Moursund, Engel, 1984) have suggested that a computer literacy or computer education course can provide introductory information and skills quite adequately, but information dealing with using the computer as a tool in teaching and learning should be integrated into existing courses, not made the subject of entirely new courses. If computer-related skills are to be incorporated into existing professional education courses, another area must be added to the ACM list of teacher training efforts—that of technology training for teacher educators. Before these skills can be infused into the teacher education curriculum, the university faculty must first develop those skills themselves.

The Louisiana Tech University College of Education Faculty realized the need for all preservice teachers to acquire computer-related skills and agreed that the preservice years are when teachers should receive that training. Of course, this required a commitment by the faculty because only one faculty member was integrating computers into a methods course in any way at that time. The faculty agreed to make a concerted effort through the technology training program to incorporate computers into their professional education courses.

Developing and implementing a plan for training teacher educators is not an easy task. Many factors affected the amount of time faculty used computers. Preliminary surveys indicated that faculty were interested not only in techniques for teaching with computers, but also in using them to generate teaching materials, support research, and support publication efforts.

DEVELOPMENT OF PHILOSOPHY TOWARDS COMPUTER USE IN EDUCATION

In the past four years in Louisiana, there have been several State initiated mandates regarding computer education. These have affected curricula, certification requirements, and course content in Louisiana high schools, colleges and universities.

1. high school graduation requirements were upgraded to include a half unit of computer literacy;
2. computer literacy teacher certification requirements were established;
3. computer literacy curriculum guides were developed and distributed;
4. computer science teacher certification requirements were established;
5. computer science curriculum guide was revised and distributed;
6. the Board of Regents of Colleges and Universities for the State of Louisiana mandated that each graduate of a Louisiana college or university must be verified to have had computer literacy training; and
7. a state adopted policy statement regarding software

copyright was established.

The above mandates focused much attention on acquisition of computer-related skills in Louisiana institutions of learning. Consequently, impetus for curriculum reform and course revisions to include computer literacy components was provided. Even those teacher educators who did not exhibit enthusiastic attitudes toward learning a new technology recognized the need to do so in response to the increasing use of computers in classrooms and the Louisiana State mandates.

Initial moves towards requiring computer competencies for preservice teachers were in the form of curriculum revision. An introductory computer education course, entitled "The Microcomputer: A Tool for the Teacher," was incorporated into elementary and secondary teacher education requirements. It is sequenced before the professional education methods courses so that students who are enrolled in methods courses will have already had the introductory computer education course.

The second step in infusing computer education into the curriculum was a commitment by all teachers of the elementary and secondary methods courses to participate in a technology training program designed to help them learn to incorporate computer-related skills into their courses. The faculty came to a consensus on the following philosophy regarding integration of computer use in the teacher education programs:

-All preservice teachers should complete an introductory course in computer education.
-All preservice teachers should receive instruction in integrating computers into content instruction in each professional education methods course.
-All teacher educators should participate in a program to develop competencies in the following areas: (1) using the computer and selecting computer software; (2) applying computers in their subject areas; and (3) using computers to develop teaching materials.
-All teacher educators should incorporate in their written course syllabi (where appropriate) activities that reinforce computer-related skills.

Once the faculty agreed on the philosophy regarding computer training, planning of the training program for the teacher education faculty began.

PROGRAM GOALS

The teacher educator technology training program at Louisiana Tech University College of Education was designed to prepare faculty to teach professional education students to use computers as a presentation and materials preparation tool for promoting learning in the content areas. Broad goals reflecting the stated philosophy of the faculty include training to enhance student and faculty:

1. Familiarity with hardware systems;
2. Familiarity with software designed for materials preparation and presentation;
3. Familiarity with types of software and its application in particular subject areas;
4. Familiarity with uses of computers in educational settings; and
5. Familiarity with criteria for evaluating software.

PROGRAM PLANNING CONSIDERATIONS

During development of a plan for training teacher educators to merge teaching and technology many factors were considered. Initially, there were practical and affective considerations to be made. Then the content was identified by establishing faculty and student competencies and developing modules of instruction for use when presenting and reinforcing computer concepts in the methods courses.

Affective Considerations

When planning a training program for teacher educators, affective considerations are very important. Participants in this program were surveyed regarding their attitudes towards computers, towards computers in instruction, and towards computers as a tool. Areas of concern exhibited were:

Motivation:

As is true in any instructional setting, providing appealing, appropriate and relevant activities was very important. Research indicates that university level teachers, when offered inservice opportunities, overwhelmingly chose topics in their own disciplines (Richardson, Fish, Okun, 1983). When surveyed, participants in this program marked items indicating a higher degree of interest and use of software that applied directly to their area of expertise or that met a particular need in their discipline. When developing the instructional activities, the format must be flexible enough to fit the needs of the particular area. The incentives that participants rated most favorably were: (1) software that was particularly applicable to their field; (2) utility programs they could use to perform a particular task relevant to their teaching; and (3) the aid given in identifying research and publication potential for the instructional activities that were developed.

Fear of computers:

A second affective consideration was alleviation of fear. A number of the participants marked items indicating a fear of computers or a fear that their colleagues knew more about computers than they knew. Much care must be taken to induce a relaxed attitude during the early training phase. Teachers and most adults have a pronounced need to save face. Errors are the horrors of many adult human beings. Every effort to turn the computer into a "warm fuzzy" rather than a "cold prickly" should be made. The first experience in front of their students should be a positive one for the students and the faculty member. Nothing can be so reinforcing as a good teaching experience with computer -- nothing so frustrating and depressing as a computer disaster. Participants should be encouraged to accept assistance in the first experiences to decrease the probability of a bad first experience for the faculty member and students.

Involvement in planning:

Evaluations indicated that teacher educators who participated in the development of the philosophy, the decision-making concerning inclusion of computer-related activities in course syllabi, and spent more time personally developing the presentations for their courses felt more successful; felt more comfortable with the teaching materials, the hardware, and the software used; and expressed more positive attitudes towards computers.

Convenience:

An important affective factor in making the program work was scheduling meetings, training sessions and tutoring sessions at times that were convenient for faculty. To accommodate the time schedules of the faculty, sessions on the same topic were often held more than once. The lab assistant, graduate assistant, and computer coordinator also aided faculty by setting up large-screen projection systems in classrooms for demonstrations; constructing learning centers; developing transparencies or handouts; assisting with making presentations in the classroom or lab when requested.

Accessibility of hardware/lab:

One of the foremost reasons indicated in the initial surveys for limited use of computer was lack of access to computers both personally and as a class. With limited numbers of computers in one lab that serves 12 sections of courses, time in the lab was often difficult to arrange. We used demonstrations with large-screen projection systems on many occasions, and set lab time each afternoon for individual use and assignments.

Expectations of supervisors:

Aside from the State mandates, other supervisory support was given to this effort. The preliminary meeting was attended by the area coordinator and revisions of course syllabi were extended to require inclusion of computer skills and uses.

Practical Considerations

Leadership:

An inservice program of this nature cannot be successful without leadership. An individual or committee must plan the program and take responsibility for carrying it out. If directed by committee, there still must be a person who is ultimately responsible for the success of the program. This program was led by the computer coordinator and supported by the Area Coordinator and Dean of the College. Committees were used to give input on particular aspects of the program and its implementation.

Grouping strategies:

Initial training on how to use the hardware and software was carried out with groups determined by the time when faculty could attend. Follow-up meetings were most often on a individual basis or with two to four faculty members who teach in related subject areas (example: reading methods, reading practicum, language arts methods). Tutoring and planning and developing materials were most often done individually.

Budget:

Our computer lab does not have a separate budget, therefore there were no specific monies committed for consultants, training, hardware, software, maintenance, or supplies to support the project.

Software and Hardware Availability:

Over a period of three years the computer coordinator acquired software to support instruction in each subject area for a range of grade levels through gift and grant funds, Teacher Education general budget, or College of Education general budget. Purchase of applications software for materials generation was given high priority.

Hardware purchases maintained compatibility among machines and included some portable machines and a large screen projection system. The University supports a Technical Services Department which repairs and maintains campus equipment at no charge if the materials needed for repair do not exceed $12.00. Repairs over that amount must be paid by the budget area. Supplies come from the Teacher Education and College of Education budgets. We are currently seeking funding to hire experts for elementary and secondary levels in each subject area to direct workshops for our faculty and other teacher educators on using computers in the content areas.

Types of programs:

Since funds were limited, training was done by the computer coordinator and lab assistants. There is often concern expressed about graduate assistants or student lab assistants teaching faculty. The preliminary meeting and initial training was led by the computer coordinator with graduate assistant and lab attendant assisting. The graduate student and lab attendant were given specific responsibilities during the programs which demonstrated their expertise in using particular software packages. Thereafter, during individual sessions or class presentations, faculty members showed no reluctance in seeking assistance from the student helpers. Evaluation forms reflected no concern regarding use of student help in training.

Number of participants:

Training groups of teacher educators that teach in the same content area would allow a workshop leader to work with a larger group of participants at once. However, since the focus of this project was to train our own College of Education personnel, we limited the first group of faculty to those who teach the methods courses: math, science, social studies, language arts, and reading for elementary and secondary levels. These courses encompassed a total of eight participants, but the difficulty of the training effort was compounded by the wide variety of grade levels and subject areas to be addressed.

Attendance:

The negative feelings generated when a training program is "required" must be weighed against the possibility of having low attendance when a training program is "voluntary." This program was not required, however, by committee decision, faculty were required to include computer activities in their course syllabi (where appropriate). All methods teachers (except one) helped to develop the philosophy of the program, therefore all had somewhat committed themselves to the project "voluntarily." To accommodate problems with scheduling, training sessions were taught more than once, and most of the follow-up and tutoring work was done individually.

Content Considerations

Developing computer-related skills among COE faculty:

Competencies for teacher educators related directly to the five goals of the program providing instruction "about computers as well as training relating to teaching "with" computers.

-Hardware manipulation - Teacher educators will demonstrate familiarity with available computer hardware and presentation equipment.
-Knowledge of materials generation software - Teacher educators will demonstrate familiarity with materials generation software by preparing a module of materials for incorporating computer-related skills in their methods course.
-Knowledge of software related to content area - Teacher educators will demonstrate familiarity with available software for use in their particular subject area(s).
-Knowledge of uses of computers in instructional settings - Teacher educators will demonstrate familiarity with ways that computers can be used in whole group, small group, or individual learning environments.
-Knowledge of criteria for evaluating software - Teacher educators will demonstrate familiarity with software evaluation criteria.

Incorporating computer-related skills into courses:

Follow-up meetings were designed to help faculty identify particular computer activities and applications that support course objectives and the enhancement of computer-related skills acquisition. The following group of student competencies , adapted from Rogers, Moursund, and Engel (1984), provided a basis for activities planned in the series of methods courses:

-Learn to operate a computer and run a computer program
-Evaluate computer-based learning materials
-Know ways to incorporate computer-based activities into the lesson and unit plan structure
-Know sources for information about software in the subject area
-Review and use a variety of computer assisted instructional software programs
-Review and use a variety of materials generation and presentation aids software
-Know related peripherals and presentation devices
-Know vocabulary and basic system hardware components
-Know a word processing system
-Know the roles of computers and implementation strategies relative to one's subject areas
-Know the potential impact of computers on society and educational processes

A consideration when implementing this program with the students was that most students had not yet had the introductory computing course. Therefore, more responsibility was placed on the methods course teacher to introduce students to the hardware system and students could not be expected to meet all competencies in the early stages of this project because of lack of training from an introductory course.

PROGRAM IMPLEMENTATION

Implementation of the training program was guided by the above considerations and objectives. The following is the sequence of events for implementation of the program.

Development of Hardware, Software and Personnel Resources

To begin the program, hardware, software, and personnel resources had to be developed. Over a period of three years adequate lab hardware to serve 30 students and a large screen projector were acquired; software in each subject area was collected; and a graduate assistant and lab assistant were employed. Area curriculum committee, area coordinator, Dean, and State mandates supported the computer education effort therefore underscoring the credibility of a training program for teacher education faculty. Although no specific budget was allocated to the computer lab or computer education effort itself, administrators gave high priority to these requests for monetary support. Without these commitments, this training program could not have been successful.

Faculty Training

Preliminary meetings were held to familiarize the teacher educators with resources, facilities, and assistance as well as with hardware, software, and teaching strategies relating to incorporating computer-related competencies in the classroom. Objectives for the preliminary meetings were:

-To distribute lists of software holdings organized by subject area - (including title, subject area, grade levels, description)
-To distribute information on ways computers can be used in the classroom, types of instructional applications for each subject area sample lesson plans incorporating software, competencies for preservice teachers, suggested activities and assignments for in-class and out-of-class reinforcement of computer-related skills.
-To distribute lab time schedule and check-out procedures.
-To familiarize participants with hardware and software holdings and evaluation criteria for each.
-To familiarize faculty with facilities, policies for use, and services provided in the center
-To familiarize faculty with classroom applications and materials preparation applications
-To review and revise lists of student competencies
-To familiarize faculty with software available in the lab for use in their areas
-To encourage faculty to request help in developing materials for use in presenting computer-related skills, in acquiring new software for their area, in learning how to use software.

Follow-up meetings were designed for individual faculty (or small groups of faculty in related areas) to meet with graduate assistant, lab assistant, and computer coordinator to discuss plans for incorporating computer-related skills into their methods courses. Objectives for the follow-up meetings were:

-To provide assistance in developing modules of instruction complete with transparency masters, handouts, student homework or assignment sheets, and any other visuals requested for use in each methods course for

developing computer-related student competencies.
-To provide assistance in identifying reinforcement assignments for students to complete in lab
-To plan schedule for implementation of the teaching module, i.e. schedule use of lab, equipment, software.

Daily hours that lab assistant, graduate assistant, or computer coordinator were available to tutor faculty on particular software packages or hardware functions were scheduled. Objectives for tutoring sessions were:

-To provide assistance to faculty on use of software
-To provide assistance with hardware for demonstrations or presentations
-To provide assistance in further development of teaching materials or student lab assignments

Methods of incorporating computers into preservice instruction

Faculty members chose several ways to incorporate computers into their methods courses. Several used the computer to develop some of their own presentation materials.

The secondary mathematics, elementary science, elementary social studies, and reading faculty members chose to have students come to the computer lab and view a short demonstration of software relating to their areas, discuss uses of such software in the classroom, and then review at least three software packages. The students submitted written reviews on a software review form that the faculty developed at one of the group meetings. It was a simple form focusing on the purpose of the software, how it could be used in a classroom setting, and students' comments regarding strengths or weaknesses of the software.

Elementary mathematics and secondary social studies faculty carried the above activity a step further, by providing an entire class period for demonstration and discussion using the large screen projection system and/or individual computers. Students were then assigned to complete software reviews outside class time. The students came in during lab hours and completed their reviews without having restrictions on time.

The elementary language arts faculty paired lab school students with methods course students and had them develop personalized stationary with holiday motifs using utility software. Then a BASIC program was used to print handwriting lines on the stationary so that lab school students could write a story with the holiday theme. Secondary language arts faculty demonstrated a word processor by loading a theme to edit, discussing what changes were needed, making the changes, and printing the edited document.

The secondary science teacher invited experts in the area of science from the university science faculty, to demonstrate software and talk about its use in science instruction. Discussions of software reviewed or demonstrated and possible uses of that software instruction were follow-up activities in each methods course.

Project Evaluation

Evaluation of the technology training program included completion of an entrance and an exit instrument by teacher educators and students that provided information on the cognitive, affective, and behavioral components of the training program. An open-ended questionnaire collected information on how the individual teacher education faculty implemented the program with their students, the response of the students and faculty to the computer activities, and future plans of the faculty for incorporating computer-based activities in their methods courses.

SUMMARY OF RESULTS

Evaluation of the information collected from faculty and students indicate successful implementation of this program. There has been a significant increase in the number of hours of computer use by faculty and students, a significant increase in knowledge of computer-related skills and information as assessed by the the exit instrument, and higher rankings of attitudes towards computers by faculty and students that at the beginning of the project.

The amount a student learns or progresses toward a stated goal is often used to measure the success of a teacher. Students in each methods course completed 15-item pre and post questionnaires designed to assess knowledge, feelings, and skills in the area of instructional computing. Of the 15 responses, five items measured cognitive information, five measured affective information, and five measured behaviors. Mean scores on each item were computed for pre and post evaluation and combined means of all cognitive items, affective items, and behavior items were analyzed. Table I. shows the mean scores for pre and post evaluations and the *t* score with probability for each item.

For each of the items 1, 2, 3, 4, 5, 11, 13, 14, and 15 the difference between pre and post scores was statistically significant. The *t* statistic was significant at $p < .0001$. For items 6, 10, and 12 the difference between pre and post scores was statistically significant at $p < .05$ level. It was noted that the three items that did not show significant change from pre to post test were items on which students scored high on the pretest. Therefore gains were not significant between the pre and the post evaluations.

When items were grouped according to cognitive, affective, and behavior measures, the difference between pre and post test scores was statistically significant for each group. The *t* statistic for cognitive measures was $t, 74 = -5.99$, $p < .0001$; the *t* statistic for affective measures was $t, 74 = -6.78$, $p < .0001$; and the *t* statistic for behavioral measures was $t, 74 = -8.95$, $p < .0001$. The difference between mean scores for pre and post evaluations of these groups indicate statistically significant gains in cognitive, affective, and behavior measures relating to instructional computing skills.

The above statistics suggest that significant gains can be made cognitively, affectively, and behaviorally by investing even one class meeting in computer training for preservice teachers. Of course, one class meeting is not the ultimate goal—and faculty indicated their wishes to extend their computer-based activities to address more computer-related skills in their content instruction. But, there is clear evidence that even small steps can yield significant results.

BIBLIOGRAPHY

Blurton, C. (1986). California's new computer requirements for teachers. ICCE SIG Bulletin. 3 (2&3), 55-58.

Poirot, J., Taylor, R., and Powell, J. Teacher education. Topics: Computer Education for Elementary and Secondary Schools. ACM. New York, 1981.

Rogers, J., Moursund, D., and Engel, G. (1984). Preparing Precollege Teachers for the Computer Age. Communications of the ACM, 195-200.

Richardson, R.C., Jr., Fisk, E. E. & Okun, M.A. (1983). Literacy in the open-access college. San Francisco: Jossey-Bass.

TABLE I.

Item	N	Mean Pre	N	Mean Post	t score	probability
1. I feel comfortable running packaged software.	45	2.96	30	6.63	-6.3929	.0001
2. I feel that I can use packaged software to make teaching materials.	46	2.96	30	6.53	-6.3274	.0001
3. I feel that I know some ways apply computer software in the classroom.	46	2.70	30	6.83	-7.7409	.0001
4. I know how to turn on the computer and run a piece of software.	45	3.73	30	8.27	-8.4363	.0001
5. I know about some software that is available for this subject area.	46	1.74	29	6.31	-7.8788	.0001
6. I feel somewhat nervous about using a computer.	44	4.55	30	2.93	2.3188	.0232
7. I feel that other students know more about computers than I do.	46	5.74	30	4.93	1.3019	.1970
8. I think computers can be useful as a tool in the classroom.	46	7.33	29	7.72	-0.9184	.3614
9. I think computers can be useful for making teaching materials in my classroom	46	7.26	30	7.47	-0.4506	.6536
10. I think I will use the computer in my classroom if one is available.	46	6.74	30	7.93	-2.7427	.0076
11. I have used the computer to view software	46	1.54	30	7.30	-8.8097	.0001
12. I have used the computer to make teaching materials	46	0.72	30	2.50	-2.7986	.0076
13. I have viewed software and discussed its use in the classroom.	46	1.59	30	6.70	-8.6874	.0001
14. I can evaluate a software package.	46	1.96	30	6.30	-6.2059	.0001
15. I can incorporate a computer activity into a lesson plan.	45	2.56	30	7.00	-6.9516	.0001

SPECIAL NEEDS/FACULTY, STUDENT APPLICATIONS

Christy Schmitt
Donald J. Slater, Barbara C. Kosloski
Cary D. Wintz, W.M. Akalou

ABSTRACT: Data Base: A Vehicle to Teach Thinking Skills To Students With Learning Disabilities

Christy Schmitt, Area Education Agency 7 Multi-Categorical Resource Room Teacher, Hudson High School, Hudson, IA

This presentation will focus on a project developed for students in a high school multi-categorical resource room. However, the structure of this project is such that the lessons can easily be converted for use in the regular classroom and for use in other subject areas besides History.

The data base was used as a teaching tool for the the secondary Learning Disabilities student. Some of the basic needs of a resource room student are: organization, structure, repetition, individualized and/or small group instruction, concrete examples, instruction that reinforces the student's learning style, and activities that build self-esteem. Using a data base as a teaching tool in the resource room meets each one of these needs.

The academic objectives for this project are that the students will: (1) develop higher level thinking skills through manipulation of their data base, (2) create a data base for their regular class (Iowa History), (3) learn how to find and how to use different types of resources (human, printed, and audio visual), and (4) learn how to operate the Appleworks data base program.

The behavioral objectives for the class are that the students will: (1) learn to work cooperatively and (2) demonstrate the use of positive reinforcement skills with each other.

The affective objective is that the students will improve their self-confidence.

The lesson plans for this project will be shared and discussed during the presentation. Each lesson is designed to help the students develop higher level thinking skills be utilizing Bloom's Taxonomy of Cognitive Objectives. The facilitator encourages higher level thinking by applying questioning techniques with an emphasis on analysis, synthesis, and evaluation questions.

ABSTRACT: A Computer for Every Student

Donald J. Slater, Computer Coordinator,
Barbara C. Kosloski, Assistant Computer Coordinator,
Sewickley Academy, Sewickley, PA 15143

What would happen to a school curriculum if all the students had access to a computer, both in school and at home? Sewickley Academy, a private day-school in the suburbs of Pittsburgh, Pennsylvania, is in the second year of a program designed to answer that question.

This year, all sophomores and juniors are expected to have access to a computer at home and in school for extensive use in almost every subject. Next year, all Senior School students will be using computers routinely, for homework as well as classwork.

All sophomores must demonstrate fundamental skills in typing and the use of AppleWorks, or they take a class that will give them these skills. In English, the process approach to writing takes full advantage of the capacities of word processing for revising, editing, and printing finished documents. Our students send and receive writing through an interstate telecommunications network, and we edit a national anthology of student work.

In history, students learn to "DO" history, through the use of simulations, databases and spreadsheets. In science, they use statistical software and a spreadsheet for calculating and tabulating the results of experiments, and through simulations they are able to extend dramatically the limits of the high school lab.

The use of the BASIC language in math classes promotes the understanding of algorithms, and the speed of the computer enables students to explore the patterns that appear in recurrent calculations.

Those who can afford to buy computers have willingly agreed to do so, and families who need financial aid are able to borrow computers from the school. Although the program is not hardware or software specific, which allows students to use computers that their families may already have at home, the Apple IIc computer and AppleWorks software are important components of our program.

The Sewickley Academy computer center is set up to provide maximum flexibility for our program: class groups can receive simultaneous instruction on individual computers; students may have access to individual computers; students who bring their own computers from home will have a place to set them up; or units from the computer lab can be moved to other areas of the school to meet a particular teacher's needs.

ABSTRACT: Faculty Development Workshop on Personal Computers

Dr. Cary D. Wintz, Asst. Director, Faculty Instruction and Support, Douglass Institute, Texas Southern University, and Dr. W.M. Akalou, Head of the Department of History and Geography, Texas Southern University, Houston, TX

As Texas Southern University began a program designed to computerize the campus by placing personal computers and/or terminals on the desk of every faculty member, it recognized the need to provide training for faculty in the use of personal computers and personal computer software. Consequently in the summer of 1987 we instituted a series of four week workshops for faculty and staff. The purpose of these workshops was:

1. To enhance faculty productivity by developing faculty computer skills.
2. To insure that the personal computer hardware and software that the University acquired would be effectively utilized by faculty and staff.

Our approach was based on the premise that the acquisition of new technology (in this case personal computers) must be accompanied by extensive efforts to this technology--to familiarize them with the operation of the technology, to reduce or eliminate their anxiety about the new technology, and to demonstrate the benefits of the new technology in a way that is meaningful to them.

The approach that we took at Texas Southern University was very effective for two main reasons. First, and perhaps most important, the director(s) of the workshop were social science/humanities faculty who were almost entirely self-taught in the use of PCs, could communicate with the faculty in non-technical language, and could, by their own example, demonstrate that one does not need to be a computer scientist or an engineer in order to fully undersatand and use Pcs. The workshop was also effective because it focused on computer programs and applications that were of immediate use and benefit to the faculty: fundamental DOS operations, word processing (Wordstar), data base managemend (dBase III) for student records and bibliographical work, and Lotus 123 for class grade books, and departmental (and other) budgets.

The effectiveness of the workshops was measured by several factors:

1. Voluntary attendance in a summer afternoon workshop by thirty-one facluty members was, in itself, almost unbelievable.
2. Feedback from participants was extremely positive.
3. Almost half of the participants in the summer workshop attended an intermediate level workshop the following fall.
4. Word-of-mouth advertising by original participants has created such a demand for the workshops that they have been institutionalized by the University, and faculty still face waiting lists to get into the workshops, even though these workshops are conducted at what is usually considered an undesireable time--late in the afternoon.
5. The workshops have created strong demand for personal computers. Faculty who initially seemed ambivalent about this new technology, now impatiently pressure their deans and the academic vice president for their own units.

MIDDLE SCHOOL

Janet Parker, Susan Brown
Brenda M. Raker
Marian Rosen
Anthony Sassi, Richard Alan Smith

ABSTRACT: Project ACT (Activities with Computer Tools)

Janet Parker, Professor, University of Louisville, Louisville, KY 40292; and Susan Brown, Codirector, Project ACT

Project ACT uses computers to address problems shared by many middle schools:

- Lack of experience and opportunity for content area teachers to use computers in teaching their regular classes;
- Low student valuing of education reflected by high dropout rate and low rate of choice of academic classes; and
- Lack of ongoing support for teachers learning new technology.

Funded by a $52,000 Innovation Grant, Project ACT is a two-year collaborative effort between the University of Louisville and Bullitt County Schools to increase student interest in schooling by having them use computers as a tool in their regular classes. Project activities, correlated to state curriculum guides, are designed to mirror society's use of computers in order to show students how the skills they are learning in school relate to future careers. For example, activities involve students using the word processor like journalists writing newspaper articles, computer probes like scientists collecting and analyzing real-time data, databases like social scientists finding patterns in historical data, and spreadsheets like statisticians analyzing financial situations. In addition, each activity is curriculum based rather than enrichment; topics chosen for activities are identified as important but difficult to teach and appropriate for approaching with computer tools.

In sum, Project ACT has developed and implemented curriculum materials for all four middle school content areas (mathematics, science, social studies, and language arts) that:

1. Make appropriate use of computer tools (word processor, databases, and spreadsheets);
2. Are correlated directly to the existing curriculum;
3. Illustrate how computers are used in various career areas;
4. Are interesting and challenging for adolescent students; and
5. Develop critical thinking skills, not merely factual learning.

In addition to the curriculum based computer tool activities and the increased student interest in classwork, other benefits include:

1. Opportunity for subject area teachers to update and extend regular textbook topics with computer applications;
2. Students beginning to develop their own problems which often involve more complex tasks and provide motivation to learn new skills; and
3. Opportunity for teachers from different schools to share materials, problems, and strategies.

This presentation will focus on:

1. Activity examples from each content area;
2. Initial and ongoing teacher training; and
3. Implementation programs and solutions.

Project ACT materials use *AppleWorks* and are not available commercially.

ABSTRACT: Electronic Spreadsheets in the Middle School Curriculum

Brenda M. Raker, Marketing Support Representative, Apple Computer, Inc. Indianapolis, IN

The computer has been described as having the potential to establish a mode of exploration, experimentation and risk-taking for the classroom, an environment regarded as integral to intellectual learning. This presentation will focus on how spreadsheets might be used with the computer to establish a problem solving environment in the mathematics classroom.

The presentation is the result of several years of work with teachers at a regional computer training center, as well as coursework at the University of Louisville. Emphasis will be placed on:

1. Using spreadsheets to enhance the

mathematics curriculum;
2. Introducing students to spreadsheets;
3. Encouraging problem solving with spreadsheets;
4. Ideas for templates for the middle school class; and
5. Classroom management techniques.

ABSTRACT: The Idea Lab

Marian Rosen, St. Louis, MO

Ladue Junior High students used LogoWriter to author software. The project was designed to demystify the process by which software is produced, to provide a library of tutorials and games written by children for children, and to meet the following objectives:

1. Challenge students' ability to define a problem.
2. Encourage divergent thinking by having students formulate multiple ways in which they can present their material.
3. Encourage convergent thinking by having students evaluate their approaches and decide on a final form for their game.
4. Involve students in two of Polya's problem-solving methods.
   a. Breaking a large problem down into sub-problems that are simpler to solve.
   b. Starting with the known and looking for ways to apply that information to the problem at hand.
5. Introduce students to the principles of interactive programming and of software design.
6. Provide an opportunity to work with Logo, which is an ideal medium for practicing Polya's methods of modular problem solving; for encouraging multiple solutions, and demanding evaluation of solutions.
7. Develop communication skills in that students were asked to write documentation for their software package.

The results of this project are a set of interactive games which we call The Idea Lab. They include tutorials on the skeleton, human organs, and early explorers. Math games include addition and multiplication drills as well as a lesson on mirror symmetry. Language games include drills on homonyms, synonyms, missing medial vowels, riddles, comprehension, and differentiating between b,d,p, and q. A musical game tests the counting and attention skills of listeners. There are also "open-enders" that draw mazes, do single-stroke Logo with some interesting twists, and allow young children to build pictures by stamping shapes.

The games have been distributed to the grade schools in our district where individual students use them during lab time. The multiplication drill is used by special district teachers at the junior high. The Ladue Early Childhood Center uses them as part of their computer curriculum for the extended day kindergarten program. The Idea Lab's greatest value, however, is as a model of Logo work that goes beyond beginning graphics.

The presentation will show selected games, outline the process by which students were guided to set such complex goals, and discuss the conditions necessary for success.

ABSTRACT: Designing a School of the Future - Guidelines for Administrators

Anthony Sassi
Richard Alan Smith, Ph.D., Houston Independent School District, Houston, TX

A description of a step-by-step approach to integrating computers into a large urban middle school. It is based upon the experience gained from the implementation of the School of the Future Project, a cooperative development effort between the Houston Independent School District and Apple Computer, Inc. The objective of the project is to infuse computers into a middle school curriculum and discover the differences they make in the educational process. The project began in 1985 with the loan of 200 Apple IIe's and 10 Macintosh computers to F. M. Black Middle School and is currently in full operation.

F. M. Black Middle School is the site of a cooperative development effort between the Houston Independent School District (HISD) and Apple Computer, Inc. This project is known as the School of the Future. Its purpose is to infuse computers into the curriculum and discover the differences they make in the educational process. The project began in November of 1985 with the arrival of 200 Apple IIe's and 10 Macintosh computers to the school building.

The use of computers was strictly voluntary. The project emphasized a grass-roots approach, and it had the complete support of the school's teacher technologist and principal. A full-time coordinator was also assigned to the school. HISD's Department of Technology provided support in the form of hardware, software, and personnel.

Distribution of Hardware and Software--Equitable distribution was essential. The project had been introduced to the teachers in the spring of 1985. The teachers who were the most interested in computers were identified at that time and became the core group of users. The majority of software supported the reading and mathematics curricula. Other software supported science and social studies.

Staff Motivation--The core group of teachers was trained first. They represented the major subject areas and were perceived as leaders in the school.

Training--Training took place mostly during teachers' preparation times and emphasized the following subjects:

- Hands-on
- Word processing: personal and instructional applications
- Data base: personal and instructional applications
- Computer-assisted instruction: for each major content area
- Classroom management
- Lab management
- How to use technology for a better teacher appraisal

The training was made as flexible as possible.

General Instructional Approach--Prepackaged commerical CAI software, which had been district approved and teacher selected, initially was used for instruction. As students and teachers became more comfortable with operating computers, applications-based instruction was implemented coupled with innovative, teacher-created lessons and materials. These ideas were shared amongst all of the teachers. The sharing of resources began to involve neighboring schools, as the information acquired was disseminated to other schools in the district via conferences, reports, and video.

THE POWER OF APPLEWORKS

Lee Hayward
Resources for AppleWorks
Cherry Creek Schools

ABSTRACT

AppleWorks is the best selling program for the Apple II line of computers. It's ease of use, the integration, and the power of each section have made it popular in the schools. Third grade through college students are using AppleWorks for their stories and writings, while their teachers are handling their chores of recordkeeping and writings on the same program.

This session, shared with Jim Carlisle, will look at some of the power available to AppleWorks users. I will present a variety of template disks that are available commercially to help teachers. This will include commercially available materials that can be used with students in Grades 3-12. Most subject areas will be highlighted with these materials.

There are also a lot of books, videos and cassette tapes that are available to help teach AppleWorks. We'll look at some that can be used as references for AppleWorks users, and some that teachers can use to teach AppleWorks to students of all ages.

I will end the session with a look at the third-party software that enhances AppleWorks, that is, spelling checkers, graphing programs, utility programs, and so forth. I'll place special emphasis on Beagle Bros' new Time-Out Series of add-ons. Time will be provided for questions and answers at the end of the session.

APPLEWORKS GOES TO SCHOOL

James Carlisle
President/Coordinator
Teachers' Idea and Information Exchange

ABSTRACT

Teachers from the elementary level to graduate schools and from Maine to Hawaii are discovering the power and potential of *AppleWorks*. Application of *AppleWorks* in the educational setting fall into two broad categories. These categories are classroom/administrative management applications and direct teaching applications. Within each of these two categories the individual uses of *AppleWorks* would seem to be as numerous and diverse as the number of individuals using the program.

The focus of this presentation will not only be on the various applications of *AppleWorks* in the educational setting, but will also touch on the way teachers and administrators are sharing these applications and building upon the work of others through the Teachers' Idea & Information Exchange.

The Teachers' Idea & Information Exchange is a nationwide "users group" for educators using *AppleWorks*. This presentation will provide educators with information on how they may become involved with the exchange and the various services it offers. The *AppleWorks* templates and files discussed during this presentation will be derived from the current collection of *AppleWorks* data disks available through TI&IE.

Each of the component parts of *AppleWorks* will be examined through demonstration and discussion of several *AppleWorks* files represntative of the kinds of applications being developed and used by teachers across the country. These will include:

1. Spreadsheet files which demonstrate the use of *AppleWorks* for constructing both simple and relatively complex gradebooks. The use of the spreadsheet for the construction of calendars, schedules, and on screen tests and quizzes. The use of the spreadsheet as a problem solving tool for students.
2. Database files which provide models for the creation of test data banks, and on screen study aids. Additional database files will be viewed which will provide a brief discussion and demonstration of some of the limitations of information stored in this way and how

students may be taught about these limitations.

3. Word processor files will be examined which provide creative writing activities for students and the production of teaching materials such as tests and lesson plans. The use of the mailmerge function in creating high interest writing exercises will also be presented.

As teachers at all levels move from teaching 'about computers' to using computers as tools, additional educational applications of *AppleWorks* will be developed. Teachers using *AppleWorks* will continue to share these applications and build upon the work of others.

BEYOND INSERVICE TRAINING - TOWARDS PROFESSIONAL DEVELOPMENT

Daniel H. Watt and Molly Watt
Education Development Center
Newton, MA

Henry Olds
Learningways, Inc.
Cambridge, MA

ABSTRACT

During the past few years a new consensus has emerged with regard to the professional development of practicing teachers. The Carnegie and Holmes reports have stressed the need for ongoing professional development throughout a teacher's career, which includes opportunities to contribute to the field as well as to learn from the work of others. In addition, recent research indicates those elements of professional development programs which are most likely to lead to improvement in classroom practice.

This panel session will summarize recent research on professional development and describe three model professional development programs in educational computing. Each model provides teachers with collaborative opportunities to contribute new knowledge to the field, to design curricula or develop software that can be shared with other teachers and students. Each program provides a focus on learning and on creating effective learning environments, that goes far beyond the specific skills needed to use a particular programming language or application tool.

Teachers as researchers investigating student learning. This model will describe an NSF-sponsored collaborative research project conducted by Education Development Center. Teachers met in a bi-weekly research seminar to investigate Logo learning among their students. In the process they learned the methods of qualitative research, identified important aspects of Logo learning to observe, defined their own research questions, collected and analyzed data, and prepared reports for their colleagues.

Teachers as curriculum developers. This model will describe the development of "Immigrants" an integrated language arts, social science and mathematics curriculum based on the use of AppleWorks word processing, database, and spreadsheet tools. This project was carried out by Henry Olds and a group of teachers at Harvard's Educational Technology Center.

Teachers as microworld designers. This model will describe some experiences at the Logo Institute--a summer Logo immersion program--in which teachers have collaborated to create Logo-based exploratory learning environments for particular subject areas.

COMPUTER SCIENCE CURRICULUM: SECONDARY

Beverly J. Sangston, Darlene Grantham, Nancy Sanchez
Wanda Kertzman
Larry E. Liden
Steven H. Schwartz, D.N. Perkins, John Kruidenier, David Niguidula

ABSTRACT: Networking - A Sharing of Resources for Computer Science and Business Education

Beverly J. Sangston, Director
Darlene Grantham, Teacher Specialist
Nancy Sanchez, Teacher Specialist
Computer-Related Instruction, Montgomery County Public Schools,
Rockville, MD

Three senior high schools in Montgomery County, Maryland, are involved in the pilot of networked hardware and software in business education and advanced computer science classes. Equipment to support each network site consists of 15 IBM PS/2 Model 30 computers, an AT file server, and two printers. The operating system for the network is Novell.

Software selection for the network labs began in the spring of 1987. Computer science and business education teachers were involved in the process of identifiying appropriate software. Final selections were based on the needs of the instructional program, usage in the business community, availability of network versions of the software, and cost.

The courses involved in the 1987-88 network pilot include: Pascal, Advanced Placement Computer Science, Typewriting 2, Accounting, and Office Practice. During the summer of 1987, a selected group of teachers developed curriculum and support materials for these courses for the MS-DOS hardware.

Teacher training, an on-going activity, began in August, 1987, with a three-day workshop on hardware, software, and supporting materials. Staff from the Department of Computer-Related Instruction visit schools to provide additional training, conduct demonstration lessons, and assist teachers in solving problems that have occured. Monthly support meetings are held after school to share classroom experiences and provide further training. Credit inservice courses, approved by the state of Maryland, are being developed to support the network applications.

As a result of the pilot, issues being addressed include: the instructional advantages of networking, teacher training needs, software licensing, lab modification, and cost effectiveness.

School system plans include phasing comparable labs into all senior high schools over the next five years, with six additional labs planned for fall, 1988. Current plans also involve the expansion of the pilot to additional courses for the 1988-89 school year.

ABSTRACT: Student Programming of Real-World Applications

Wanda Kertzman, Computer Science Instructor, Washington High School, Tulsa, OK

This curriculum project has been developed for an advanced computer programming class at the high school level. It is the result of trying to meet the needs of several types of students: Those whose prior experience or talent has made them regard regular class assignments as being too trivial; those who long for a challenge and must see the relevance of what they do in a concrete form; and those who see no need to structure their work in accordance with accepted programming practices.

By encouraging students to participate in the development of software having a practical use in the school, the community, or for an individual, the problems mentioned above can be resolved. In addition, several other goals can be realized:

1. Students will learn to program creatively but efficiently as they search for solutions to real problems.
2. They will develop communication, critical thinking, and logic skills as they work with 'clients' and fellow programmers to complete a project.
3. While the students gain experience they will also be performing a service for others.
4. Students will be motivated as they receive a strong and practical background for continuing their studies at the college level.

This technique can be used for either individual or group assignments. Programs of varying complexity have been used with both first-year and advanced students, illustrating that with careful structuring, prerequisites may be kept to a minimum. The projects can be integrated with regular coursework, and are a viable solution for the teacher faced with varying levels of ability in a single class.

Organizational strategies have been developed to aid the classroom teacher in coordinating several simultaneous projects. A certain amount of planning for individualized lessons will be needed for students who are working independently, and the teacher must also be ready to cope with student frustrations and failures at times.

Success rates and amount of output will vary from year to year, depending on student abilities. The important rewards, however, will include: student understanding of the value of structed programming techniques; student interest in the study of advanced computer science topics; a heightened sensitivity to ethics issues; the cooperation and growth of students who have worked together on a program; and an increase in student self-confidence and pride.

ABSTRACT: The Advanced Placement Computer Science Curriculum Project

Larry E. Liden, Superintendent, the Institute of Computer Technology, Sunnyvale, California

The Institute of Computer Technology, a public school agency specializing in facilitating education about technology, was funded by the California legislature to develop an advanced placement computer science curriculum guide. The guide was developed by teams of educators and computer scientists from various silicon valley industries who spent over 3,000 man hours developing and testing lesson plans, overhead transparencies, lab exercises, worksheets and handouts.

To involve as broad a range of experts as possible, the curriculum was developed in modules. Using the course description produced by the College Board, five individual modules were identified. Teams of practicing computer programmers and computer science teachers were assigned to each module. Estimates of required times were modified to fit a 150 hour maximum so that the course could be taught in a school year with time for review before the AP test in early May. Teams worked with curriculum development specialists to develop materials to cover the content within the prescribed time frame.

The modules were tested first with a small group of six students. After revisions, they were tested again with a regular class taught by one of the developers. The individual modules were then merged into a two part course and tested again. In addition, teams of technical reviewers from local companies reviewed the materials for relevancy and accuracy.

Evaluation at each stage was based on written feedback from teacher and student and on student performance. The final evaluation was, of course, student performance on the College Board AP test. At every stage where the test has been administered, at least 50% of the students have earned the highest possible score.

The materials are now available and being used by schools throughout the nation. However, revision continues as the course undergoes testing at two ICT sites and feedback is received from others.

The project has been exciting. The interaction between industry and education has been rewarding. The extensive revision and testing process was impressive. The materials are receiving good reports and, hopefully,their availability will encourage more schools to offer advanced computer science.

The process has since been used to develop a course on microcomputers as a productivity tool for college and work is underway on a course in C Language.

Copies of all materials are available for the cost of duplication.

ABSTRACT: Developing a Metacourse for BASIC.

Steven H. Schwartz, D. N. Perkins, John Kruidenier, David Niguidula.
Educational Technology Center, Harvard Graduate School of Education, 337 Gutman Library, Cambridge MA. 02138

The programming group at ETC has conducted both clinical and experimental research for a number of years on how elementary and high school students learn to program (in LOGO or BASIC), and identified a number of difficulties common to many novice programmers. These will be discussed, along with a "metacourse" developed to provide mental models, problem-solving strategies, key concepts, and other structures that may help students understand more deeply and wield more artfully the knowledge they aquire during their regular instruction in BASIC. These materials are designed to be integrated into the teacher's normal course materials as an enriching "vitamin shot" rather than a substitute for the usual course content, and hence require little additional teacher training.

Highly encouraging results, in terms of increased mastery of BASIC, found in two large scale empirical studies conducted in a number of high school BASIC programming classes will be detailed. While teachers were also quite positive in the evaluation of the metacourse, many however indicated that they wanted more freedom than provided in the tightly scripted conditions of the metacourse "lessons" . Thus we have revised the metacourse into a modular "packet " format which we are in the process of testing.

This current Metacourse consists of three types of "packets" or modules which: a)*introduce* some fundamental concept or strategy, b) help the teacher *infuse* these concepts or strategies throughout their course, or c) provide specific ways in which such concepts or strategies can be *transferred* to other programming languages, academic subjects or problems in real life. What has become apparent as we have worked with teachers is that many of them share a goal of teaching their students -- "What computers and programming languages are really all about", "how they can be used as problem-solving tools"-- and not just the syntax of a particular programming language like BASIC. We believe that a powerful pedagogy of programming which directs attention to mental models, strategies, and other heuristics, may lead to more success at achieving both programming competence and these broader problem-solving skills.

INFLUENCING LEGISLATION: IMPLICATIONS OF THE OTA REPORT ON EDUCATIONAL TECHNOLOGY

Anne H. Knight, *Chair*
President, NH Association for Computer
Education Statewide (NHACES)
New Hampshire Technical Institute
Concord, NH

ABSTRACT

This panel will focus on legislative activity at the state and national level related to technology in education. The impact of a national study on educational technology will be assessed. The panelists represent various domains of education and are familiar with the policy making process and how to influence legislation.

The Office of Technology Assessment (OTA) of the United States Congress has released the results of a study called *Educational Technology: An Assessment of Practice and Potential*. This report is likely to serve as a guide for the formation of policy on educational computing at the state and national levels. Because of its importance, the report should be studied carefully and evaluated in terms of underlying assumptions about education, computing, and society.

Featured in this panel will be an overview of the OTA report. Dr. Linda Roberts, who directed the project, will discuss the background and intended goals of the report as well as describe the research done by OTA. Panelists will comment on the report's implications for the future of computer technology in schools and colleges. The ICCE sees this panel as an important forum for assessing the impact of the report.

The focus of the panel will shift to national and state legislation, passed and pending, on technology in education. Methods of influencing legislators and other policy makers will be described. California and New Hampshire will be used as examples of both effective and ineffective means of influencing legislation.

The ICCE strives to facilitate the transformation of education by working with other groups (such as ACM/SIGCAS) to influence the creation and implementation of effective educational policy related to technology. This panel is designed to inform and broaden the perspectives of people on policy formation and legislative activities related to educational computing.

PANELISTS

Linda G. Roberts
Office of Technology Assessment
United States Congress
Washington, DC

James B. Stedman
Library of Congress - CRS/EPW
Washington, DC

Charles L. Blaschke
Education TURNKEY Electronic Distribution
Falls Church, VA

Ronald E. Anderson
University of Minnesota
Minneapolis, MN

LeRoy Finkel
San Mateo County Office of Education
Redwood City, CA

SPONSOR: ICCE and ACM/SIGCAS

ELEMENTARY MATH AND SCIENCE

Virginia Morrison
Albert Ortiz
Denise A. Schmidt, Linda G. Smith

ABSTRACT: Improving Skills in Story Problems Through Word Processing Activities

Virginia Morrison, Educational Computer Consultant, Bay City Texas ISD

Development of problem solving skills is a continuing concern for educators in our world of rapid information expansion. Through word processing editing a student can identify pertinent information in a math story problem and change some of the information to reflect their own ideas within a structured mathematical sentence. The sense of ownership motivates students to think about the situation and solve the problem.

Four classes in our elementary schools began using story problem activities in word processing during the Spring of 1987. Story problem lessons were written in word processing files to correlate to appropriate grade level curriculum guides and to TEAMS[1] skill objectives. Students progress from a beginning activity of finding numbers they can change to locating the object words in the problem which they replace with their own image words. At this point, they often realize that the problem's question no longer refers to their own images and editing lets them correct an "error" in the story problem they are developing. Excitement is strong because a student is controlling information in an area of math that has often been difficult.

This year we have expanded the number of classes using the word processing story problems. In the computer labs students take their turns editing a problem or writing a new problem from the lesson model. Printed copies of their own story problems are used by the classes to share and solve. Calculators have been introduced in the lab to encourage problem solving thinking.

Various themes and skill levels used in the story problems integrate into an overall sequential plan. Several themes such as numbers in your school room, holidays, menus or measurement are modeled at each skill level. Editing or writing your own story problems calls attention to specific information, however teachers need to transfer understanding the information into organizing and implementing that knowledge. Using your own words offers an opportunity to begin drawing pictures or symbols to aid development of strategies.

Student motivation is high as they work with their own ideas using model story problems. They want to think of the largest number or their favorite kind of food. A positive attitude toward story problems is growing in these classes. Teachers are gaining insight into building problem solving strategies by bridging from the student's known world into the written word presenting new situations.

ABSTRACT: Mergence of Software in Archaeology with Fourth-Grade Students

Albert Ortiz, Project Archaeologist and Third-Grade Teacher, Desert View Elementary School, Gadsden I.S.D., Sunland Park, NM 88063

An archaeological survey and excavation located at Desert View Elementary School in New Mexico allowed students to participate in archaeological methods of identification and assessment with the assistance of the computer. Computer analysis of the artifacts indicate a variety of subsistence-related activities and exploitation of the environment around the site, which dates to about 1100 A.D..

This field season marked a dramatic advance over earlier work. It employed (a) the practical skills and theoretical perspective of the project archaeologist; (b) software in quantitative analysis to address the objectives of archaeological methodologies in the interpretation of archaeological resource; and (c) the abilities of a well-trained fourth-grade Bilingual crew that is proficient on Apple IIe, IBM PC, PC and PS/2 Model 30 computers.

Utilizing nontraditional methods of computerized instruction, students recognized new computerized strategies being used in this project:

1. Logo's indispensable element of a graphic approach was used to teach the students the art and science of plotting points on the archaeological site with the aid of an Engineer's Transit. Plane geometry and plane trigonometry were used for computation when using the transit. Logo provided sufficient

---

[1] Texas Educational Assessment of Minumum Skills

knowledge for locating points and orientation of lines based upon measurement of angles and lengths for gathering and processing information about the physical environment within the archaeological site;

2. An archaeological database was created using application software (*RapidFile* and *ChartMaster*). This enabled crew members to do an interpretation of recovered material by means of quantitative analysis; and

3. Participants were permitted to decide what the data enabled them to conclude about the archaeological resource of the area (especially the subsistence data which is lacking for prehistoric human social systems along the Rio Grande of the physiographic region of West Texas El Paso and South Central New Mexico).

ABSTRACT: Integrating the Computer into the Curriculum and Other Secrets

Denise A. Schmidt and Linda G. Smith, Self-contained classroom teachers, Independence Community School District, Independence, IA

Formalizing the teaching of thinking has become a concern nation-wide. It has been stressed nationally in the area of mathematics. At the state level, Iowa has included both teaching technology and teaching thinking skills in new state standards to be implemented by 1992. Locally, the Independence School District has put emphasis in these areas. As second grade teachers, we have woven technology and the teaching of thinking skills into our math curriculum.

Using one computer, connected to a large monitor and directly using a piece of software with a room of second graders, a stimulating, learning, thought provoking environment was created. By offering a variety of experiences, students are better able to develop mathematical concepts. Pre-computer activities which used manipulatives were designed. The handling and involvement with these concrete objects helped to prepare the students for computer interaction. The entire class solved problems presented by the computer program: Gertrude's Secrets. The purpose of these activities was to further help students transfer information gained in one experience to new situations. Specific questioning techniques were carefully modeled and students developed their own questioning strategies. Students worked in pairs practicing their questioning techniques. This helped them to internalize the self-questioning process.

Through this project further professional staff development was created and implemented. In-services were conducted to model the process which was developed for other educators in the Independence School District and throughout Iowa.

The computer has been a catalyst for involving our students, teachers, and parents. It gave new energy to our use of thinking skills and questioning strategies. We all became excited about the learning process.

AN EFFORT OF THE ICCE TELECOMMUNICATIONS INITIATIVES

G. Christopher Clark, *Chair*
Delaware-Chenango BOCES
Norwick, NY

ABSTRACT

An effort of the ICCE Telecommunications Initiative, the session will begin with an overview of technologies now in use for distance learning around the United States and Canada. Educators involved with ongoing programs will then explain how they use technologies like satellite service, audiographic conferencing, facsimile transmission, interactive television, and electronic messaging.

Telelearning Project - G. Christopher Clark

The phone rings and a voice says, "I'm going to put you on the bridge and we'll get started." That's not the sound of a civil engineer talking to his foreman; instead, a class has just started at one of seven area high schools. Using regular telephone lines to combine very small groups of students at two or more schools, the Delaware-Chenango BOCES (Board of Cooperative Educational Services) Telelearning Program is able to offer opportunities which were previously not cost-effective.

Students at Afton, Bainbridge-Guilford, Hancock, Mount Upton, New Berlin, Otselic Valley, and Oxford are participating in five advanced courses: Calculus, Chemistry, English, European History, and Music Theory. At the end of the year many will have college credit and/or placement credentials to reward their work. The concept of long-distance learning has been around for at least twenty-five years, but audiographic teleconferencing, the technique being used here is a relatively new development. Sites are equipped with a telephone handset, a computer, and a graphics tablet, allowing each participant to send and receive text, data, pictures, and voice communications. The *bridge* mentioned above is a service which connects three to forty parties in one telephone call. It has allowed the joining of five different schools in last year's Advanced Placement Calculus course.

The main goal of the program is to provide rural children with cost-effective connections to the world around them. "Electronic Field Trips", where classrooms make phone calls to musicians, authors, distant classrooms, and politicians are also arranged.

Learning by Satellite - Leigh Walters

Many rural schools face a serious shortage of secondary school teachers, particularly in mathematics, science, and foreign languages. When Oklahoma educators recognized the need to improve curricular offerings to students in that state's rural areas, they formed a partnership to address the problem. These educators-from Oklahoma State University's College of Arts and Sciences, the Oklahoma State Department of Education, and from throughout the state-saw a solution in satellite broadcast. With financial assistance from the Oklahoma legislature and contributions of many satellite dishes from Cotton Rural Electric Services, these educators created the Arts and Sciences Teleconferencing Service.

The project successfully combines educational satellite teleconferencing, computer-assisted instruction, and personal contact with high school students. The service began broadcasting German I from the university's educational television studio in the fall of 1985. By the next fall, word of the program's success had spread: the number of subscribers doubled. Today, ASTS serves approximately 2000 students who take German I, German II, advanced placement physics, advanced placement calculus, and trigonometry at sites in Oklahoma, Colorado, Arizona, Arkansas, Kansas, Missouri, Tennessee, Montana, Louisiana, Washington, Wisconsin, North Dakota, and West Virginia.

Audiographics - Dennis Wydra

Dennis Wydra will be discussing audiographics. The title of his presentation is "An Effective and Economic Approach to Teleteaching."

The panel will conclude with a summary of the growth and direction of distance learning in the United States.

PANELISTS

Gary Hasloff
Texas Education Agency
Austin, TX

Leigh Walters
Oklahoma State University
Stillwater, OK

Dennis Wydra
Mansfield State University
Mansfield, PA

SPONSOR: ICCE

ISSUES IN COMPUTER SCIENCE ETHICS: A SURVEY OF STUDENT VIEWS

Laura Marie Leventhal and David Warner Chilson
Department of Computer Science
Bowling Green State University
Bowling Green, Ohio 43403

[This work was supported by the Computer Science Department, Bowling Green State University. Acknowledgements: The authors would like to extend their appreciation to Wade Gibson and Roy Johnson for their assistance in the preparation of this paper.]

Abstract

A survey of attitudes, among college computer science students, toward a variety of computer-related ethical issues was conducted. The purpose of the study was to examine relationships between computer science education, gender, and ethical issues. The survey involved the use of several scenarios; students rated the extent to which several possible courses of action were ethical or unethical. A factor analysis revealed that students have three patterns of responses, including: justification on the basis of unawareness of others, justifcation on the basis of business issues, and justification on the basis of "doing the right thing." Subsequent analysis of these factors suggested that students at higher computer science education levels tend to emphasize "doing the right thing", while students with less education respond more favorably to the other types of justifications. Female students tended to respond more favorably to "doing the right thing" than male students. That the ethical responses of students tend to change with education level suggests that computer science educators, directly or indirectly, have an influence on these responses. Educators are challenged to not only continue to make this influence positive, but also to strengthen it.

Introduction

Parker, in an ACM self-assessment procedure dealing with ethics in computing, states that:

> The need for special ethical considerations in the computer field arises from several unique characteristics of computers and their use. . . Unlike the computer field, other sciences and professions have had hundreds of years in which to develop ethical concepts that form the basis for dealing with new issues. . . an ethical community is maintained in each discipline as these codes are transferred from professor to student in universities. In contrast, computer science and technology have been in existence for only 30 years. (Parker, 1982, edited by Weiss).

While Parker's statement was directed at the computer science community as a whole, it raises many important and special questions for computer science educators. In particular:

> "How do computer science students view the special ethical considerations in computer science?" and "what type of ethical community is passed from computer science educators to students?"

Attitudes toward ethical issues and their development, outside of computer science, have long been the object of debate and systematic study (cf. Piaget, 1932; Kohlberg, 1981). But, as Parker has suggested in the previous quotation, focus within the computer science community on issues of ethics is a relatively recent phenomena. For example, in a landmark study, Parker (1981) explored attitudes toward computer science ethical issues among computer science professionals. Notably missing from the literature, however, are empirical studies which focus on the attitudes of students or beginning computer scientists.

To address this gap, we recently conducted a survey which tapped attitudes of computer science students toward ethical issues in computer science. The purpose of the study was to identify patterns of ethical concerns among this group, and to explore relationships between levels of computer science education and these patterns of concerns. That is, our study was intended to address the two questions raised above. Moreover, some ethics researchers, outside of computer science, have suggested that gender may play a role in ethical responses in general (cf. Gilligan, 1982). To explore this issue, we also considered the effect of gender among our student sample on patterns of ethical responses.

Method

Materials

An ethics survey instrument was developed. The instrument consisted of a series of demographic questions, followed by a set of scenarios. The demographic items consisted of both open-ended and closed-ended questions concerning computer science education, gender, and other background information. The scenarios described hypothetical computer science situations. Two of the scenarios were drawn from business settings; these scenarios concerned software marketing and software contracting. The other two situations were drawn from college settings; these scenarios involved a friend in need of software and a programming student. The software marketing scenario was a modification of Scenario 2.5 - "Programmer, Business Enterprise: Selling a Program No Longer Used" from Parker (1981). The other three scenarios were based on our own personal experiences and observations.

Following each of the scenarios was a set of possible courses of action for the principal player in the scenarios. Figures 1 - 4 show the scenarios and the associated courses of action.

In each of the scenarios, subjects were instructed to "rate each action in terms of how ethical you feel it is" and to "consider each action independently." The items were rated on a seven-point scale, ranging from 1 = "very unethical," 2="unethical," 3="somewhat unethical," 4="no feeling one way or the other," 5="somewhat ethical," 6="ethical," 7="very ethical." At the end of the set of courses of actions, the subjects were instructed to answer an open-ended question: " Thinking back on the actions, what was the single most important issue, in your opinion?"

Subjects

Two-hundred thirty-nine students in computer science classes at Bowling Green State University completed the survey. The subjects were drawn from undergraduate computer science courses at all levels, ranging from freshman-level introductory programming courses to senior-level advanced topics courses.

Sixty percent of the subjects were male and forty percent were female. Twenty-nine percent of the subjects had declared computer science majors; the subjects were drawn from a total of fifty-nine different majors. The computer science educational level of the subjects ranged from a minimum of no completed computer science courses to a maximum of fifteen completed computer science courses, with a mean of 2.52 completed computer science courses. The distribution of completed computer science courses is shown in Figure 5.

Procedure

The survey was distributed to subjects during the Spring Semester, 1987 by the instructors of the courses included in the study. Some instructors allowed the subjects to complete the survey in class, while others asked the subjects to complete the task outside of class. All subjects were given extra credit for completing the task in their respective classes.

Results

Across all of the subjects, the mean rating on the ethics items was 3.03. The item rated as most ethical was in response to the Friend in Need of Software Scenario. In the action, the friend is loaned the software on a part-time basis. As a conseqence of the part-time loan, the friend flunks out of school. The item rated as least ethical was involved with the Programming Student Scenario. In the action, the programming student steals the working program of a fellow student, makes slight modifications, and turns in the slightly-modified work as his own. The mean responses to these actions were 5.05 and 1.65, respectively.

In the open-ended questions, where subjects were asked to identify the single most important issue in their opinion, several patterns of responses occured. For the Software Marketing Scenario, the most common types of responses were no response (21%) and personal betterment (21%.) For the Friend in Need of Software Scenario, the most common types of responses were one copy should not matter to the company (26%), no response (21%), and piracy (19%). For the Programming Student Scenario, the most common types of responses were stealing and plagiarism (29%), copying someone else's work (22%), and no response (21%). For the Software Contracting Scenario, the most common type of response was no response (36%).

Factor Analysis

A factor analysis of the responses to the twenty possible actions was performed. The purpose of this analysis was to categorize the actions, on the basis of subject responses, into a new and smaller set of variables. These variables represent patterns or categories of of actions, rather than single actions. A principal components extraction, followed by a varimax rotation was performed. (Descriptions of factor analysis and of its use as a data reduction technique can be found in Child, 1973 and Harman, 1976.)

Three factors emerged from the analysis, using a minimum loading criteria of + or -.40. Eighteen actions will be included in the discussion of the factors. Two actions will not be included in the discussion because they either did not load with a magnitude of at least .40 (+ or -) on a defined factor or because they loaded onto two factors.

"No One Will Notice" Action Factor - The first factor included eight actions. In each of these, the action was justified, in the statement, by a suggestion that "no one will notice the action anyway." In particular, one of the actions involved the theft of a roommate's software package, while the roommate is away for the weekend. Six of the actions involved the theft and modification of another students program. One of the actions involved underbiding on a government contract. In each case, the player performed an action which would not necessarily be known by anyone else. The actions and loadings are listed in Table 1.

"That's Business" Action Factor - The second factor included courses of action which were implicitly justified by business motivations, such as profits and contested ownership of software. Four of the actions involved a decision about marketing software outside of the company in which it was developed. These actions involved the ownership rights of the creator of the software, as well as the marketing strategies of the original company. The other two actions involved actions of a manager in a software contracting firm. In each case, the manager took actions which apparently would increase his firm's profits but were misleading to the customer. The actions for this factor are listed, along with their loadings, in Table 2.

"The Right Thing to Do" Action Factor - The third factor included items which in which the principal player did the "right thing" in a conventional sense. In each of the courses of action, "doing the right thing" led to a painful consequence. For example, one of the courses of action is in response to the Friend in Need of Software Scenario. In the action, the owner of the software counsels the friend to study hard and to do her best. As a consequence, the friend flunks out of school. The four elements in this factor are listed in Table 3, along with their loadings.

One of the items in this factor had a negative loading, in contrast to the other loadings in the factor. The difference in loading direction implies that this action is at the other extreme of this factor, relative to the other actions. Close inspection of this course of action reveals that in fact it specifies "doing the wrong thing" in a conventional sense. The outcome of "doing the wrong thing" is a happy outcome. It is not at all surprising that this action was loaded in the opposite direction of the elements in the "Doing the Right Thing" Factor.

For each of the factors, factor means were calculated for each individual. The relationships between formal coursework in computer science, gender, and the three factors were investigated using multivariate and univariate analyses of covariance. (Kirk, 1968, Namboodiri, Carter, and Blalock,1975, and Norusis, 1985 provide an overview of these analyses techniques.) In these analyses, number of completed computer science courses and gender were used as independent variables and the factor means were treated as dependent variables.

The results of the multivariate analysis showed that both the number of completed computer science courses and gender had a significant effect on the three Action Factors (F(2,234) = 14.51, p < .0001 and F(2,234) = 4.42, p < .004, respectively). These results suggest that having taken a greater number of computer science courses and being male or female is significantly related to a student's responses to courses of actions involving the ethical issues included in the instrument.

Univariate analyses of the three individual Action Factors were also performed. In these analyses, number of completed computer science courses and gender were again treated as independent variables; the three factors were treated as dependent variables. Analysis of the "No One Will Notice" Action Factor showed that number of completed computer science courses was a highly significant predictor (F(1,236) = 22.49, p < .0001). Similarly, analysis of the "That's Business" Action Factor revealed that number of completed computer science courses was a strong and significant predictor (F(1,236) = 22.47, p < .0001). For the "The Right Thing to Do" Action Factor , both gender and number of completed computer science courses were significant predictors (F(1, 237) = 14.11, p<.0002 and F(1, 236) =24.27, p<.0001, respectively).

The relationship between the number of completed computer science courses and responses to the first two factors wqs negative. That is, increasing numbers of courses led to decreased ethical ratings for the "No One Will Notice" and the "That's Business" Action Factors. In other words, subjects who had taken more computer science courses decreasingly rated these forms of justifications as ethical than subjects with fewer courses in their background.

By contrast, the relationship between "The Right Thing to Do" Action Factor and the number of completed computer science courses was positive. Subjects who had completed a larger number of courses also tended to emphasize "doing the right thing" in their responses.

The mean scores of the male and female subjects for the "The Right Thing to Do" Action Factor were 4.13 and 4.61, respectively. In other words, females were more likely to give a higher response to the actions in the factors than males.

Conclusions

In our introduction, two questions were raised:

> "How do computer science students view the special ethical considerations in computer science?" and "what type of ethical community is passed from computer science educators to students?"

The results of our survey address both of these questions, at least for the types of issues which were raised in our instrument. In response to the first question, our results suggest that computer science students experience at least three main types of ethical considerations. Specifically, the considerations which emerge from this group involve awareness and unawareness of others, business issues, and "doing the right thing", in a conventional sense. Moreover, in terms of ethical considerations, students are not a homogeneous group, but in fact show some differences based on gender. Relative to the second question, the experience of taking computer science courses, and presumably receiving some influence from computer science educators, does lead to an altered perspective of ethical issues. In particular, students with higher education levels appear to be less tolerant of actions which are justified by unawareness of others or business issues. These students seem to favor conventionally moral actions to a greater degree than students at lower levels of education.

For computer science educators, these results are especially noteworthy. In particular, they suggest that computer science education and computer science educators do have a strong impact on students in this arena. Moreover, the results suggest that this impact is positive! As educators we are challenged to make our influence even more positive. We are challenged to provide a comprehensive ethics educational experience, both in terms of classroom study and practical models of behavior.

References

Child, D. (1973) The Essentials of Factor Analysis. New York: Holt, Rinehart, and Winston.

Gilligan, C. (1982) In a Different Voice: Psychological Theory and Women's Development. Cambridge, MA: Harvard University Press.

Harman, H.H. (1976) Modern Factor Analysis. 3rd ed. Chicago: University of Chicago Press.

Kirk, R.E. (1968) Experimental Design: Procedures for the Behavioral Sciences. Belmont, CA: Brooks/Cole Publishing Company.

Kohlberg, L. (1981) The Philosophy of Moral Development. San Francisco: Harper and Row.

Namboodiri, N.K., L.E. Carter, and H.M. Blalock. (1975) Applied Multivariate Analysis and Experimental Design. New York: McGraw-Hill Book Company.

Norusis, M.J. (1985) SPSSX: Advanced Statistics Guide. New York: McGraw-Hill Book Company.

Parker, D.B. (1981) Ethical Conflicts in Computer Science and Technology. AFIPS Press.

Parker, D., edited by Weiss, E.A. (1982) Self Assessment Procedure IX: A self-assessment procedure dealing with ethics in computing. Communications of the ACM. 25. 181-195.

Piaget, J. (1932; 1965) The Moral Judgement of the Child.
New York: The Free Press.

Scenario I: A.J. is an employee of the XYZ Corporation. She and several of her coworkers developed the Z+ Software System for use by the XYZ Corporation. XYZ used the software for awhile but now the company no longer needs the system. Through some contacts outside of XYZ, A.J. has learned that there is a possible market for the Z+ System outside of XYZ. That is there are customers who would be willing to purchase the Z+ System, as well as support for the system.

Courses of Action

Action I-1 A.J. presents the idea of outside sales of Z+ to several managers within the XYZ corporation. However, since XYZ is not specifically a software company, no one inside of XYZ is interested in marketing the product. Even though A.J. believes that Z+ would be very successful, she decides to just forget about marketing Z+ at the current time. She reasons that the company probably knows best anyway.

Action I-2 A.J. presents the idea of outside sales of Z+ to several managers within the XYZ corporation. However, since XYZ is not specifically a software company, no one inside of XYZ is interested in marketing the product. A.J. knows that this is a great product and decides to market Z+ independently of the XYZ Corporation. However, being a little short on capital, She decides to keep hers/his job at XYZ and just market Z+ on the side. For fear of being fired, A.J. does not tell anyone at XYZ about marketing Z+ on the side.

Action I-3 A.J. presents the idea of outside sales of Z+ to several managers within the XYZ corporation. However, since XYZ is not specifically a software company, no one inside of XYZ is interested in marketing the product. A.J. knows that this is a great product and decides to market Z+ independently of the XYZ Corporation. She forms her own company to market and support Z+. Of course, A.J. needs some help in this venture, so she entices other developers of Z+ away from XYZ to help with the new company.

Action I-4 A.J. feels that the Z+ product is really, at least in part, her property, since she put so much work into the development of the product. A.J. decides to try to make some money for all of her efforts so she forms her own company to market and support Z+ independently of the XYZ Corporation. A.J. changes the name of Z+ to +Z as well as some of the features so that the product is not exactly the same as the original. A.J. does not worry about stealing the product from XYZ, since it is not specifically a software company anyway and probably would not have any interest in selling the product.

Action I-5 The NFT Company is a large software company which produces and sells software similar to Z+. The addition of Z+ to their line of products would really give them an edge in this market. Without talking to anyone at XYZ, A.J. offers to give Z+ to NFT, providing that NFT hires him/her as a manager. With that new position, A.J. would be in a position to make important decisions and also make considerably more money.

A.J. does not worry about stealing the product from XYZ, since it is not specifically a software company anyway and probably would not have any interest in selling the product. Besides, A.J. feels that the Z+ product is really, at least in part, her property, since she put so much work into the development of the product.

Figure 1 - Software Marketing Scenario

Scenario II: T.L. has just purchased the powerful new software package, *Q*, for her personal computer. T.L.'s friend A.M. also owns the same type of personal computer and is impressed by the power of *Q*. Unfortunately, *Q* is much more expensive than A.M. can afford. A.M. is a good student, but for a number of reasons is about ready to flunk out of school. She recognizes that the use of *Q* would help in at least a couple of courses. A good grade in those courses would get her out of trouble.

Courses of Action

Action II-1 T.L. knows that A.M. really admires *Q* and would love to use it. However she feels that A.M. should be able to make good grades with whatever she has available. She tells A.M. to study hard during the rest of the term and do the best that she can. This advice is not enough to help A.M. and A.M. flunks out.

Action II-2 T.L. feels sorry for her friend. Seeing that the use of *Q* would really help A.M. survive this term, she loans A.M. *Q* . Of course T.L. needs *Q* for her own classes, so A.M.can not use it as much as she would like to. The part-time use of *Q* is not enough to help A.M. and A.M. flunks out.

Action II-3 T.L. feels sorry for her friend. Seeing that the use of *Q* would really help A.M. survive this term, she loans *Q* to A.M., so that A.M. can make her own copy of the software. T.L. knows that there is a copyright on *Q* but reasons that A.M.'s copying is ok. After all, T.L. did not sell the system to anyone else. Besides, without the copy, T.L. is sure that A.M. will flunk out of school. At worst, the developers of *Q* will only be out a few dollars. Sure enough, using her own copy of *Q* A.M. manages to improve her grades.

Action II-4 T.L. feels sorry for her friend. Seeing that the use of *Q* would really help A.M. survive this term, she loans *Q* to A.M., so that A.M. can make her own copy of the software. T.L. realizes that other friends could do better in school if they had *Q* also. T.L. loans out *Q* to some other friends to copy. She charges them a nominal fee of a beer and a pizza. Not surprisingly, T.L. begins to have a great social life as well as the satisfaction of seeing the grades of friends improve. T.L. knows that there is a copyright on *Q* but reasons that a few extra copies won't hurt the company that developed *Q* anyway.

Action II-5 A.M. is really desparate for *Q*. When T.L. is home for the weekend, A.M. goes into T.L.'s room and borrows *Q*. She notices that the diskette has a copyright notice but figures that one little copy won't hurt anyone. She makes a copy of *Q* and does not tell T.L. about it. When T.L. comes back A.M. says that her parents scraped together the money for *Q* and bought it. Not surprisingly, A.M.'s grades begin to improve dramatically.

Figure 2 - Friend Who Needs Software Scenario

Scenario III: J. J. is a sophomore computer science major at Bowling Green, currently taking several computer science courses, as well as several electives. J. J. has a part-time job, is active in intramural athletics, and has an "average" social life. J. J. is a good student, but for a number of reasons has fallen behind in one of his computer science courses. A major programming project in that course, which was assigned three weeks ago, is due in just one more week.

Courses of Action

Action III-1 A friend of J. J.'s offers to complete the project (i.e., write the rest of the program) for a fee. J. J. accepts the offer, and turns the completed program in on time as if it were his own work.

Action III-2 J. J. goes through the computer services distribution files late one evening and takes a program listing that belongs to another student in his class. He copies a majority of the program, and turns the completed program in on time. He reasons that he did change some parts of the logic, and also changed a number of the procedure names and data names. He concludes that the program is therefore "his own" and that he will not be caught.

Action III-3 J. J. goes through the dumpster outside the Union late one evening and finds a program listing that belongs to another student in his class. He copies a majority of the program, and turns the completed program in on time. He reasons in this case that he did not really steal the listing since it had been thrown out and was therefore "fair game" to take. J. J. copies a majority of the program, and turns the completed program in on time. He reasons that he did change some parts of the logic, and also changed a number of the procedure names and data names. He concludes that the program is therefore "his own" and that he will not be caught.

Action III-4 In a stroke of "sheer luck", J. J. finds a program listing lying on the sidewalk between the Union and the Math-Science Building. He copies a majority of the program, and turns the completed program in on time. He reasons in this case that he did not steal the listing since it had been lost and was therefore "fair game" to take. J. J. copies a majority of the program, and turns the completed program in on time. He reasons that he did change some parts of the logic, and also changed a number of the procedure names and data names. He concludes that the program is therefore "his own" and that he will not be caught.

Action III-5 In a stroke of "sheer luck", J. J. sits down at one of the PC's in the Union and discovers that someone has mistakenly left his diskette in the PC. Upon listing the contents of the diskette, J. J. finds that it belongs to one of the students in his class. He prints a copy of the program on the diskette, but does return the diskette to its proper owner, consoling himself in the knowledge that he has done a "good deed" by returning it. Using the printed listing, he copies a majority of the program, and turns the completed program in on time.

Action III-6 A good friend of J. J.'s, who took the course from the same instructor a year ago, offers to give J. J. a listing of his completed program, as well as the diskette that still has the program on it. J. J. makes a copy of the program, makes extensive "global" changes in procedure names and data names, and turns the program in on time. He reasons that he did not "take" or "steal" his friend's program, but instead was "given" the program and therefore has no reason to feel guilty about using it, especially since he changed so many names and made it look as if it were his own program.

Figure 3 - Programming Student Scenario

Scenario IV: K. K. is a project manager for ABC Software, a contracting firm that develops software systems for the government and the military.

Courses of Action

Action IV-1 K. K. has decided to underbid on a 1-year project for the government so that ABC Software will be more likely to get the contract. He knows that the project will require 5 man-years of effort and will cost approximately $800,000. He places a bid of $550,000, reasoning that when the project is about three-fourths done, ABC Software should be able to request an additional $250,000 because by that time the government will have so much money invested in the project that it will have no other choice but to "endless" amount of money to spend and that $250,000 is a "drop in the bucket".

Action IV-2 K. K. has decided to bid on a 1-year project for the government as if the work involved was being done "from scratch". In fact, ABC Software recently completed a similar project (an inventory management system) which is nearly identical to that being requested by the government. K. K. knows that the first system can be properly modified (customized) at a cost of about $300,000. The first system was completed at a cost of $1,200,000 and K. K. reasons that the government should be both willing and able to pay that full amount. After all, that is what the government would have to pay if the system were truly being developed "from scratch".

Action IV-3 ABC Software has just received a contract to develop a system for a large industrial firm. The bid stated that M. M., a senior programmer, would be the primary programmer. Since receiving the bid, K. K. has decided to put M. M. on a larger and supposedly more important project. Now N. N., a junior programmer whose salary is about three-fourths that of M. M., will be the primary programmer on the first project. K. K. has decided not to revise the cost to the industrial firm, even though ABC Software's profit for the one-year project will now be greater due to N. N.'s assignment as the primary programmer.

Action IV-4 ABC Software has just received a contract to develop a system for a large industrial firm. Since receiving the bid, K. K. has learned that another software firm has already developed a system which, with a few minor changes, would meet the large industrial firm's needs. K. K. decides not to mention the other firm's product, and instead goes ahead with development of the new system from scratch.

Figure 4 - Software Contracting Scenario

ITEM	LOADING
Action II-5 T.L borrows and copies A.M.'s copy of "Q". T.L. says that she bought "Q".	.528
Action III-1 A friend of J. J.'s completes the project for a fee, and J.J. turns the project in as if it were his own work.	.755
Action III-2 J. J. takes another student's program from the distribution files, copies a majority of the program, and turns it in as if it were his own work.	.853
Action III-3 J. J. takes another student's program from the dumpster, copies a majority of the program, and turns it in as if it were his own work.	.892
Action III-4 J.J. finds another student's program lying on the sidewalk, copies a majority of the program, and turns it in as if it were his own work.	.843
Action III-5 J.J. finds another student's diskette in a P.C., prints a program, copies the program, and turns it in as if it were his own work.	.804
Action III-6 A good friend of J. J.'s, who took the course from the same instructor a year ago, gives J.J. a program as well as his diskette. J.J. copies the program, with global changes, and turns it in as if it were his own work.	.756
Action IV-1 K. K. underbids on a 1-year project for the government so that ABC Software will be more likely to get the contract.	.521

TABLE I - "NO ONE WILL NOTICE" ACTION FACTOR

ITEM	LOADING
Action I-2 A.J. presents the idea of outside sales of Z+. Because of lack of interest within XYZ Corporation, A.J. decides to market Z+ independently of XYZ Corporation (on the side) and does not tell anyone at XYZ Corporation.	.537
Action I-3 A.J. presents the idea of outside sales of Z+. Because of lack of interest within XYZ Corporation, A.J. decides to market Z+ independently of XYZ Corporation (forming her own company and entices developers away from XYZ Corporation.	.667
Action I-4 A.J. feels that the Z+ product is really, at least in part, her property, since she put so much work into the development of the product. A.J. decides to try to make some money for all of her efforts so she forms her own company to market and support Z+ independently of the XYZ Corporation and changes the name of Z+ to +Z.	.654
Action I-5 The NFT Company is a large software company which produces and sells software similar to Z+. Without talking to anyone at XYZ, A.J. offers to give Z+ to NFT, providing that NFT hires him/her as a manager.	.525
Action IV-3 ABC Software has just received a contract to develop a system for a large industrial firm. Even though the bid stated that M.M., a senior programmer, would be the primary programmer, N.N. , a junior programmer, is assigned as the primary programmer with no revision to the cost of the project.	.553
Action IV-4 ABC Software has just received a contract to develop a system for a large industrial firm. Although another firm already has a system that would meet the firms's needs, ABC Software goes ahead with development of a new system	.512

TABLE II - "THAT'S BUSINESS" ACTION FACTOR

ITEM	LOADING
Action I-1 A.J. presents the idea of outside sales of Z+. Because of lack of interest in XYZ Corporation, A.J. decides to forget about marketing Z+ at the current time.	.530
Action II-1 T.L. knows that A.M. really admires "Q" and would love to use it but feels that A.M. should be able to make good grades on her own. A.M. flunks out.	.761
Action II-2 T.L. feels sorry for her friend. Seeing that the use of "Q" would really help A.M. survive this term, she loans A.M. "Q" . T.L. needs "Q" for her own classes, so A.M.can not use it as much as she would like to. The part-time use of "Q" is not enough to help A.M. and A.M. flunks out.	.445
Action II-3 T.L. feels sorry for her friend. Seeing that the use of "Q" would really help A.M. survive this term, she loans "Q" to A.M., so that A.M. can make her own copy of the software. Using her own copy of "Q" A.M. manages to improve her grades.	.546

TABLE III - "THE RIGHT THING TO DO" ACTION FACTOR

THE EFFECTS OF COMPUTER EXPERIENCE ON COMPUTER ANXIETY

Susan R. Wallace
Division of Computer and Information Sciences
University of North Florida
4567 St. John's Bluff Rd., S.
Jacksonville, FL 32216

ABSTRACT

This study looks at the effects that completion of a semester long computer science course had on students' computer anxiety levels. The subjects were university students enrolled in four sections of an introductory computer science class. Completion of this course satisfied a computer proficiency requirement for all majors in the college of arts and sciences and for secondary education majors in the college of education. The course content included computer history, terminology, algorithm design, problem solution and microcomputer implementation of problem solutions in BASIC. Computer anxiety was measured at the beginning of the semester before the students had any computer exposure and again during the last week of the semester. The Computer Anxiety Index of the Standardized Test of Computer Literacy was used to measure computer anxiety. The posttest computer anxiety scores of males were significantly lower than the males' pretest scores and significantly lower than the posttest scores of the females. Computer anxiety levels among males decreased but computer anxiety levels among females remained unchanged. This study suggests that the effect of completing an introductory computer science course on computer anxiety level is sex dependent.

INTRODUCTION

Computer proficiency is now an integral part of the basic skills that an educated person should have. The National Commission of Excellence in Education (1983) called computing a new basic skill and the College Board's Educational Equality Project (1983) stated that students should know how to operate a computer. Colleges and universities offer introductory courses in computer science for those who either had no chance or avoided the chance to acquire computer proficiency.

Students enrolling in these college level classes may bring with them some degree of computer anxiety. This anxiety has many causes and is shown to be related to a number of factors including fear of the unknown, fear of embarrassment or ridicule, a hesitancy to ask questions, uneasiness caused by seeing younger people seemingly so comfortable with computers and avoidance caused by an unpleasant initial experience (Widmer and Parker, 1983; Rivizzingo, 1980). Some people are overwhelmed by the sheer volume of new information and the time required to absorb it (Rubin, 1983).

In addition, relationships have been found between computer anxiety and the length of computer exposure or the amount of computer knowledge. In high school and college students, the amount of previous computer experience and the level of computer knowledge is hypothesized to be inversely related to the level of computer anxiety (Loyd and Gressard, 1984; Bitter and Davis, 1985). In other words, those students with less computer experience and knowledge were more computer anxious.

Sex differences with respect to computer anxiety have also been noted in some studies (Dambrot, et al., 1985; Vredenburg, et al., 1984; West, 1987) but not in others (Loyd and Gressard, 1984). In two of the studies where differences were found, the females had higher computer anxiety levels than the males. However, West (1987), in comparing the anxiety levels of males and females at the beginning of an introductory computer course, found that males were more anxious than females.

It may be reasonable to expect that exposure to and use of computers during an introductory course would lower computer anxiety levels by eliminating fear of the unknown and by increasing computer experience and knowledge levels. The effect of a seemingly overwhelming amount of new information should be alleviated by the structured presentation of material in a classroom setting. This hypothesis is supported by studies that measured computer anxiety levels at the beginning and end of a semester long introductory computer course (Thompson, 1982; Wresch and Hieser, 1984). Computer anxiety posttest scores were significantly lower than the pretest scores in each case. However, a brief exposure to computers through the use of CAI materials was found to be ineffective in decreasing computer anxiety (Wresch and Hieser, 1984).

THE PRESENT STUDY

This study compares computer anxiety levels of a cross section of college students at the beginning and end of a semester long required introductory computer science course. Computer anxiety was measured at the beginning and at the end of the course.

Subjects for this study were university students in four introductory computer science classes. In an attempt to minimize any effects due to the classroom teachers, the classes of two female teachers who had each taught the course during at least two previous semesters were selected for the study. All four of the classes were daytime classes. The course is the one used by liberal arts majors and secondary education majors to satisfy a computer proficiency requirement. It includes computer history, vocabulary, problem solution, algorithm design and hands-on programming in BASIC. All four classes were given an introductory demonstration on the operation of the microcomputer system they would be using for class assignments by a microcomputer lab employee. The subjects had unscheduled and unlimited access to the microcomputer lab which was open 80 hours a week. During lab hours, one student assistant was available to help with machine commands and the functioning of the hardware while a second student assistant was available to help with programming logic and program debugging.

The Computer Anxiety Index (CAIN) of the Standardized Test of Computer Literacy (Maurer & Simonson, 1984) was used as the measure of computer anxiety. The CAIN is a 26 item six-point Likert scale questionnaire. The subjects selected a response to statements such as "I enjoy using computers" and "I am usually uncomfortable when I have to use computers" ranging from "strongly agree" to "strongly disagree." Possible scores range from 26 to 156 with a higher score indicating a higher anxiety level. The CAIN, constructed and validated at Iowa State University, has an internal consistency reliability estimate of 0.94 and a test-retest estimate of 0.90. During the administration of the Computer Anxiety Index, it was renamed the Computer Opinions Survey to minimize the possibility of answer bias. It was given at the beginning of the semester before the students had any computer exposure as part of the course. The survey was administered again during the last week of classes in the semester after students had completed all course work and had written approximately eight BASIC programs.

RESULTS

One hundred and seventy-six students completed the computer anxiety pretest. There were 96 males and 80 females ranging in age from 18 to 39. All classifications (freshman through graduate student) were represented with the median classification being sophomore. The pretest anxiety scores ranged from 26 to 106. One hundred and nine students (54 males and 55 females) completed the computer anxiety posttest with scores ranging from 26 to 112. A two by two analysis of variance with sex and time as the independent variables and computer anxiety scores as the dependent variable shows a significant interaction (see Table 1).

Group means are presented in Table 2. Posteriori analysis indicated that posttest computer anxiety scores of males were significantly lower than the males' pretest computer anxiety scores and that the posttest computer anxiety scores of females were significantly higher than posttest computer anxiety scores of males (although there was no difference between males and females with respect to the pretest scores). In addition, differences occurred along both diagonals; females' pretest scores were significantly higher than males' posttest scores and females' posttest scores were significantly higher than males' pretest scores. A graphical representation of the data appears in Figure 1.

Since approximately 38 percent of the students who had taken the computer anxiety pretest had withdrawn from the course before the posttest was administered, a t-test was computed to see if differences in original computer anxiety levels were to be found between participants who finished the course and those who withdrew. These groups did not vary significantly with respect to pretest computer anxiety scores.

In order to consider the amount of change that occurred in computer anxiety levels, the difference between posttest and pretest scores was computed. Difference scores ranged from -34 to +39. Since higher scores indicate higher computer anxiety, a negative difference indicates a decrease in computer anxiety level over time.

For the group who finished the course, correlations were computed between final course grade, pretest score, posttest score, difference score, age and class. Significant positive correlations were found between age and grade ($r = 0.21434$, $df = 139$, $p = 0.0113$), class and grade ($r = 0.19024$, $df = 140$, $p = 0.0244$) and age and class ($r = 0.41965$, $df = 139$, $p = 0.0001$). Neither age nor class was significantly related to either pretest, posttest or difference scores. Posttest scores were negatively correlated with grades ($r = -0.27882$, $df = 108$, $p = 0.0035$). Those students with the least computer anxiety at the end of the course tended to make the highest grades. A negative correlation was also found between difference scores and grades ($r = -0.21203$, $df = 93$, $p = 0.0413$). Those subjects earning the highest course grades tended to have the largest decrease in computer anxiety (as shown by the largest negative difference scores).

DISCUSSION

The present study lends support to the premise that completion of a semester long introductory computer course reduces computer anxiety. However, this effect was noted only in males. At the beginning of the study, males and females did not differ with respect to computer anxiety scores nor did they differ with respect to computer experience. All subjects were computer novices (little or no prior computer knowledge). It appears that male and female computer novices are equally computer anxious and that computer knowledge and experience gained through coursework lowers computer anxiety levels only in males. This conclusion is in accord with Dambrot (1985) who reported that freshmen females were more negative towards computers than were males. However, the males in that study also had more computer related courses and a greater knowledge of computer languages.

Other studies (Wresch & Hieser, 1984; Thompson, 1982) have reported non-sex dependent decreases in computer anxiety levels for those completing a computer course. Wresch and Hieser do not report the number of males and females that participated in their study. If that group was predominantly male, the apparent difference in the results of that study and the present one might be explained. The number of males and females was not reported in the Thompson study either; however, since the course taken by subjects in the study was a course for preservice teachers, it seems reasonable to assume that the students were predominantly female. If that is the case, then the results of the Thompson study and the present study do not agree. Thompson noted that the preservice teachers were taking the computer course as an elective. In the present study, the computer course was a requirement for liberal arts majors and secondary education majors. This suggests that initial computer anxiety may be easier to overcome when it is coupled with enough interest to motivate enrollment in an elective computer course. The fact that the courses used in the two studies varied in content may also help to explain the differences in the results. The course in the Thompson study included a variety of topics such as BASIC, LOGO, word processing, spread sheets, etc. It could be that using the computer to run a number of applications packages coupled with some programming is more conducive to computer anxiety reduction than is using the computer solely for programming. This hypothesis is not supported by West

(1987) who measured computer anxiety levels of computer novices at the beginning and end of a semester long introductory computer packages course. The course in the West study included three applications packages -- a spreadsheet, a word processor and a database. It is interesting to note that the males in that study were more computer anxious than the females, but anxiety levels did not change over the course of the semester. It is also possible that the single item self-assessment question used in the Thompson study was not as accurate in measuring computer anxiety as the longer questionnaire used in this study.

This study suggests that the effect of completing an introductory computer science course on computer anxiety level is sex dependent. The experience is sufficient to significantly reduce computer anxiety levels in males but has no effect on the computer anxiety level of females. It appears that a course which uses applications packages rather than a programming language is even less effective in reducing computer anxiety. We may need to find alternative techniques that will be effective in reducing computer anxiety in females.

REFERENCES

Bitter, Gary & Davis, Shelley (1985). Measuring the development of computer literacy among teachers. *AEDS Journal*, 18(4), 243-53.

College Board (1983, May). Academic preparation for college: What students need to know and be able to do. A report of the Educational Equality Project, New York.

Dambrot, Faye H., et al. (1985). Correlates of sex differences in attitudes toward and involvement with computers. *Journal of Vocational Behavior*, 27(1), 71-86.

Loyd, Brenda & Gressard, Clarice (1984). The effects of sex, age, and computer experience on computer attitudes. *AEDS Journal*, 18(2), 67-77.

Maurer, Matthew & Simonson, Michael (1984, January). Development and validation of a measure of computer anxiety. Paper presented at the Annual Meeting of the Association for Educational Communications and Technology, Dallas, Texas.

National Commission on Excellence in Education (1983). A nation at risk: The imperative for educational reform. U.S. government printing office, Washington, D.C.

Rivizzingo, Victoria (1980). Overcoming the fear of using the computer and basic statistical methods. *Journal of Geography*, 79(7), 263-68.

Rubin, Charles (1983). Some people should be afraid of computers. *Personal Computing*, 7(8), 55-57.

Thompson, Ann (1985). Helping preservice teachers learn about computers. *Journal of Teacher Education*, 36(3), 52-54.

Vredenburg, Karel, et al. (1984, August). Sex differences in attitudes, feelings, and behaviors toward computers. Paper presented at the 92nd Annual Convention of the American Psychological Association, Toronto, Ontario.

West, Beth (1987). Computer Anxiety and Its Correlates. Unpublished thesis. University of North Florida.

Widmer, Connie & Parker, Janet (1983). Micro-anxiety: How to beat it before you get it. *Electronic Education*, 3,23-24.

Wresch, William & Hieser, Rex (1984). Computers across the curriculum. *EDUCOM Bulletin*, 19(1), 20-22.

Table 1

Source	Sum of Squares	DF	F	P
Sex	4145.71	1	11.29	0.0009
Error	69057.04	188		
Time	213.84	1	2.92	0.0908
Interaction	1657.89	1	22.64	0.0001
Error	6810.77	93		

Table 2

		Pretest	Posttest	Totals
Males	mean	58.01	51.17	55.55
	std dev	16.04	14.37	
	N	96	54	
Females	mean	62.05	64.84	63.19
	std dev	15.20	20.28	
	N	80	55	
Totals		59.84	58.06	

```
 Group    64 -|                                F
          62 -|          F
  Mean    60 -|
          58 -|          M
Anxiety   56 -|
          54 -|
 Scores    2 -|                                M
          50 -|
              .
              .
              ________________________________

                 Pretest                    Posttest
```

Figure 1.

INFORMATION LITERACY:
APPLICATION SOFTWARE AND CRITICAL THINKING
AN EXPERIENCE AND A PROPOSAL

Harriet A. Cinque
Dennis J. Santomauro
Department of Mathematics and Computer Science
Kean College of New Jersey
Morris Avenue
Union, New Jersey 07083
201-527-2666

ABSTRACT

Since microcomputers have become easier to use, we believe that the early introduction of application software packages to non-computer science majors effectively develops student productivity in his major. Accordingly, at Kean College we offered two pilot sections of such a course to management science and accounting majors after we had been awarded a New Jersey Department of Higher Education Grant in Computers in Curricula Program. This paper is divided into two parts; the first of which describes our experiences in developing and presenting the course. The second is more conceptual in nature and suggests how critical thinking skills could be developed through a restructured computer literacy course utilizing application software and suggests a future direction for general education.

PART I: THE COURSE

Background

At the college, management science and accounting majors are required to take a two-course sequence in computer science. These courses were designed to provide the business major with an introduction to the computer and an understanding of the nature of programming and the problems encountered by the programmer. The first course was designed as a broad-based computer literacy course while the second had as its major objective developing in the student both an awareness and an understanding of data and file manipulation in a business environment. This approach was appropriate for an era when computer systems within business and industry were largely centralized and individuals in areas of functional management relied on members of computer staff to provide the information necessary for planning and decision making.

However, the introduction of the microcomputer and the widespread availability of application software have changed the work place. Computer systems are smaller and have become decentralized and directly accessible to people in all functional areas of business. With this in mind, we examined the two courses.

The first course seemed appropriate whereas the content of the second did not. The former had been revised in the preceding year and was being taught in a laboratory equipped with IBM PCs. The latter focused upon the need for the business person to communicate his or her data requirements in programming terms. The language RPG was utilized and accessed on one of the college's centralized computing systems, a PRIME 9750. This second course was restructured to reflect the changes in the business world with respect to the microcomputer.

The specific objectives of the revised course center upon creating a student awareness of the importance of the microcomputer as a management tool as well as equipping the student with the knowledge to adapt readily to changes in technology in the work place. Tantamount to achieving these objectives is the need for as much student "hands-on" as possible.

Since the management science and accounting majors comprise approximately 2500 students, the largest major at the college, our primary concern was the lack of hardware. The route chosen was to apply for funding under the State of New Jersey Computers in Curricula Grant Program. We submitted a proposal to equip two microcomputer laboratories, one as a dedicated classroom and the other for student use outside the classroom. Each lab would include 15 IBM PCs and a printer and the necessary software and documentation. Also included was the request for LAN support as well as personnel. The grant was awarded and funded at $75,000. (We had requested $89,000.) Subsequently, we revised the curriculum and offered two pilot sections in the Spring semester 1987.

Course Content

We developed the course to address the role of the microcomputer "in the development and organization of data in files and databases for information generation." It accomplishes this through the study of "the selection and proper use of microcomputer application packages to fulfill the information needs of business and support management problem solutions" [18]. It acknowledges that with the rise of the end-user approach to solving problems, the emphasis has shifted from the tools used to analyze to the problems being analyzed. Hence, there is greater emphasis on the nature of the problems themselves. Since this is the second course in a year sequence, it has more depth than that of an introduction or survey. Topics include:

-a review of computer system components
-an in-depth study of the nature of data including its structure, organization, storage and operations
-concepts and functions of information systems
-categories of software
-systems and program development processes
-end-user application development
-software packages
-communications
-system integrity

-social and ethical issues.

Since we felt that the use of the 'correct' text would contribute greatly to the success of the course, we conducted an extensive search during the semester prior to the pilot. We wanted a concept approach; hence a 'generic' text. We knew also that some form of lab manual was a necessity. The texts we examined with the exception of the one finally chosen seemed to fall into three general categories for this type of course:

(1) the survey introduction to computers/information processing compendiums of a little of everything without depth in anything
(2) the cookbooks of package skills, heavy on the 'how-to' but light on the 'why?'
(3) the templates for solution of specific applications.

Since the introduction to computers fell in the realm of the college's first service course, we shied away texts in category (1); the remaining two categories were judged too limited in scope.

Late in the term, a new text was published which spanned the survey and templates categories by offering a comprehensive view of the tasks microcomputers can handle through case studies; these were embedded in a generic solution model focusing on overall concepts rather than specific packages. We selected this text: *Microcomputer Applications* by R. Grauer and P. Sugrue. Its title is perhaps misleading since it approaches the subject from the data management point of view. We are satisfied with our choice. An exit survey after the pilot sessions confirmed our view; students found it interesting, appreciated its contemporary flavor and enjoyed reading it. An aspect of the book appropriate for management science students is the review of the computer science industry through a series of corporate profiles on various companies and individuals in the field.

Classroom Laboratory

Design A conscious decision was made to integrate the classroom and laboratory as opposed to the usual operation of computer science courses which separate the two. The microcomputers were placed "galley style" along both sides of a classroom; the instructor at one end of the room can easily monitor all the screen displays with a sweeping glance. This physical layout was an advantage while conducting "package feature walkthroughs"; one could easily tell a student's progress by observing the screen in relation to all the others.

Advances and interest at the college in the area of networking prompted the decision to install the microcomputer equipment under a local area network. A token ring configuration was chosen; its requirement demanded groups of seven machines per Multiple Station Access Unit and as a result a network of 14 machines was created in the classroom. Since the same classroom was to function as an open laboratory during non-class times, it was decided to place the server and the printer in an adjoining room for security reasons. This setup limits the instructor's access to the server during class time; however, this caused no difficulty since the class was not studying networking per se.

The software packages selected for study were the full application packages used in industry not limited student versions. Having the network eliminated the problem of handling the multiple sets of floppy disks required for operation of several application packages. This factor alone justifies network use. The lab assistants found the task of controlling the distribution of only 14 diskettes (those containing network files) time consuming and frustrating. It is fair to say that diskette handling would probably be the biggest problem in the operation of a microcomputer laboratory.

Usual class size in computer science courses is 30 students. Obviously, with 14 machines available, doubling up (and even, tripling) was necessary. The ratio of two students to a machine was comfortable and even proved productive because of student interaction. We anticipated the benefit of peer interaction in advance since this has been proven a distinct learning technique [19,20]. However, more than that produced a crowded situation. In the future, we plan to limit class size to 28 and maintain the 2 student : 1 machine ratio.

Lab Installation and Management The operation of the laboratory was functionally separated from the presentation of the course, thereby 'freeing' the involved faculty from the time consuming responsibilities of lab and network management. In order to do this, knowledgeable personnel are needed in the form of a network manager and a staff of assistants. After our experience with the pilot for a semester, we are convinced that any network undertaking in the collegiate, or for that matter any, environment requires a staff dedicated only to 'servicing' the network and the physical environment housing that network.

A graduate from our own computer science program became our network 'resource person'. He had participated in a department project studying network installation and was familiar with network operations and problems. Since his experience exceeded that of the faculty involved with the course, his assistance with the design and the development of the network was invaluable. His position was made available by state funds solicited for a Program Improvement Proposal. In effect, he functioned as our network manager and had sufficient background and expertise to revise the network program. He provided a network menu which listed all the software and packages available for use in the laboratory.

Lab usage totaled 15 hours weekly: 2 hours daily open lab (10 hours) and two sections meeting twice a week for 1 1/4 hours (5 hours) in the lab as a classroom laboratory. The open lab and laboratory classroom necessitates the presence of lab assistants whose functions are to:

(a) set up, maintain and service the equipment
(b) assist students with the use of a package (but not to do their homework assignments!)
(c) monitor lab usage
(d) control distribution of diskettes interfacing with the network
(e) assist the instructor during the lab portion of the class.

Towards the end of the term prior to the pilot course, we started to set up the team of 4 student lab assistants. The ideal lab assistant would be a junior with some microcomputer background in hardware and application packages. They were difficult to find especially when other jobs are available. We finally got off the ground with some computer science students with a PC background and others with good academic credentials but very little PC background. Few had any extensive experience with the packages but all demonstrated the willingness and capacity to learn.

Our 'resource person' prepared a training program involving hardware and software. He utilized the students to set up the equipment in the lab. On the whole the lab assistant staff was dedicated and dependable and functioned very well under the auspices and training of our 'resource person'. They demonstrated an allegiance to their position and the lab. That allegiance is a necessity when students are working with students. The lesson learned was that solid tight management under a single person responsible for the entire program is required for a well-functioning student laboratory. This supports the findings of Spence and Windsor [21].

In the fall term following the pilot we have 14 sections taught by 8 instructors. A second microcomputer laboratory with equipment made available through grant funds was installed during the summer. It is now our classroom laboratory. Lab usage is obviously heavier. The original lab is now the open lab. It functions 14 hours a day and 5 hours on Saturday for a total of 75 hours a week. We again encountered the problem of a limited pool of students available as lab assistants. To remedy this, we added the students who had completed the pilot course to the pool.

Teaching Strategies

When the classroom is also a laboratory, an approach beyond the lecture technique is a necessity. Not to use the equipment during a class would be wasteful of the resources. Initial ventures alternated between a short lecture followed by a short hands-on session and then an attempt to return to the lecture format within the same class session. We soon found that once the student's attention was directed to the hands-on exercise, it was almost impossible to continue with lecture since the student's interest was diverted to the problem on the microcomputer. Accordingly, class sessions are structured to lecture first followed by a directed laboratory session under the instructor's guidance.

Since there are only 14 machines and twice that amount of students, 'teams' of students work at each machine. Each student is required to provide a work disk and bring it to each class meeting. There is a tendency for the same student in a team to be the 'keyboarder' during the class. The instructor should oversee the alternation of students at this function.

The presence of the instructor in the lab environment necessitates a varied approach from that of merely following a lab manual. It offers the opportunity of the 'guided walkthrough' where the instructor can demonstrate various package features and usage techniques to all students at once. Hence, the lab manual for the course need not be as complete or detailed as one used in a course where lab is an independent venture for the student. When working with application packages during the pilot, we employed several approaches to the lab portion of the class. This was an attempt not only to determine the most effective one but also to determine the appropriate format for the accompanying lab manual. In lab sessions, we varied the amount of explanation of the features and the extent of the laboratory directions. Students favored some explanation and demonstration prior to their efforts in the lab and preferred lab directions without verbose explanations. Since we were using an application package for word processing that was not covered by existing lab manuals and we were not satisfied with the other portions of lab manuals, we began to develop our own set of lab procedures based on these findings.

Lab manuals provided with textbook packages are another complete area of study but several points are worthy of mention here. When reviewing a lab manual for possible adoption, remember that it will be utilized in two different ways. First, it will be a guide through the initial experience with the feature being studied. Then it will also be used as a reference for subsequent use in the lab. Too often, lab manuals are extremely good at providing direction for the first use but fail miserably as a reference. The same directions which were very detailed and complete for introductory use covering each step of the way become extremely frustrating for students searching for a specific direction. Identification in the form of headings with referral 'tags', paragraph names or in-margin referral points are a necessity. A lack of defined referral points cannot be compensated for by merely including a list of commands or directions in the back of the manual.

Another point to be considered is whether the particular package should be taught functionally (how to use it) or conceptually (the form of the package and what the package can do). With management science and accounting majors, the functional approach appears appropriate in the beginning. After some experience with package features in general, a combined approach may prove acceptable. Time constraints did not permit further investigation during the pilot.

It should be noted that as the students finished one package and proceeded to study the next, the time involved with introduction decreased. The instructor could make use of the features of the earlier packages studied as reference points for comparison. The more packages studied, the more they became familiar with the features employed by all packages to achieve a successful user interface. They had acquired sufficient knowledge and experience to become critical of package features toward the end of the course. Indeed, this was one of the goals of the course.

As with any computer science course, the students were assessed on a combination of tests and laboratory assignments to determine their grade. Various modes of testing were employed from traditional to hands-on. The subject of testing should be reevaluated as well as the mode of testing for this course. The traditional test (pencil to paper) is definitely appropriate for evaluating theoretical type questions. However, since a fair amount of learning in computer science is experiential, so also testing should be partially experiential. Many times a student cannot immediately provide an exact answer to a question addressing the specific use of a package on a traditional test. But having the microcomputer and package available for use allows him the opportunity to 'evoke' the answer. For example, in the pilot section on word processing, a portion of a test was conducted on the microcomputer and consisted of "hands-on" revision of a document provided by the instructor. It was not a lengthy exercise but it truly measured the student's ability and mastery of the tasks involved. The results of a student's efforts were obtained by printing/viewing the document on his work disk. This is a lengthy and time consuming task for the instructor. More research must be done on automating testing systems which can appropriately measure deviation from the desired results of package use.

We evaluated the pilot and, as a result, formulated several future enhancements. The demonstration of an on-line information search would give firsthand experience with communications. The study of business graphics and information presentation techniques will be extended. The use of videotapes and on-line tutorials will be investigated. Another evaluation for improvements will follow after the first offering of the course to all students.

Observations

Throughout the pilot offering, we were cognizant of student activity during class as well as lab. It was a way of evaluating student reaction to the course and informative in indicating areas of concern. Some of our observations follow. It is generally what we expected, although there was some variance at times.

-Some students assimilated the material very quickly and moved on at their own faster pace. Prior exposure to PC usage was not necessarily a factor.

-Initially there was very little response to information requests made by the instructor. This was somewhat surprising since some of the material was review from a prerequisite course. We found a mixed background since some students had taken the prerequisite in high school or at another college some time ago. This may also have been due to the fact that students were not expecting this revised course.

-There was difficulty with association skills. An example was not associated with a concept until they actually did it.

-Analysis and synthesis of information had to be directed when working with information stored in a database or playing 'what if' with a spreadsheet. It was obvious that students did not have much experience in this area.

-Creativity raised doubt and required reassurance. Students stumbled over minor things such as filename selection unsure of what was required and had to be encouraged to 'play' with the packages to get a feel for what they were capable of doing. Once encouraged to act freely, they were more relaxed about working with the packages.

-Student satisfaction with the course was generally high and there was an obvious level of interest which is frequently lacking in required courses. The students felt they were learning something useful. Soon into the first major topic, word processing, students began to do work for other courses and during the spring break, several students came in to work on term papers.

-There was a significant amount of peer interaction. Students willingly helped each other and did so before requesting help from the instructor or from a lab assistant.

-The atmosphere in the open lab was positive overall. An esprit de corps developed in groups of students and progressed to the entire room.

-Students experienced a sense of accomplishment when they were able to design and execute a solution to a problem. We were surprised that their sense of self-satisfaction, at times, was independent of the correctness of their solution. Using the software to solve a problem seemed as important as, if not more important than, the results.

-Confidence in use built as usage increased but not at the same level for all students.

-It was obvious not only to the instructor but also to fellow students which students had neglected to 'practice' in the lab.

-Student frustration was evident at different times. The student would know what he wanted to do and was searching through a copious amount of text to find it. A higher level of frustration occurred when a student did not know "where to start" to solve a problem.

-Students were recommending the course to others and commenting "everybody should take this course". A few students who had already received a grade for the pre-revised course requested permission to register for the revised version as a free elective.

PART II: THE PROPOSAL

The Computer Literacy Question

There have been numerous articles written concerning the concept of computer literacy. Everyone realizes its importance and believes it should be an integral component of the educational process. However, it is a concept that seems to elude defining (this should not seem surprising since computer science, in general, eludes definition [2]) and, as such finds very diverse and varied implementations. Some universities, such as Carnegie-Mellon, have implemented a computing skills requirement for all students [9].

One definition for computer literacy that appears to be on the mark is from Anderson [1]: "Computer literacy is best defined as whatever computer knowledge and skills one needs to function effectively in a given role. This includes the ability to evaluate appropriate uses of the computer, to plan and execute various applications of the computer, and most importantly, the ability to understand how computers are impacting us socially, psychologically, culturally, and ethically."

It is apparent in the literature [3,4,5] as well as the plethora of textbooks focusing upon microcomputers that the computer knowledge and skills one needs to function effectively center upon application software rather than programming. Many colleges and universities have implemented such a course and we at Kean have done the same. Although our course has a dedicated population, we believe that the content material could be restructured and presented in a manner that would develop critical thinking skills for all students, irrespective of their major. We propose such a course be required for students in all curricula and that it should fall under the rubric of general education. To support our proposal we examined some of the literature in both education and computer science.

Rationale

A strong component of all collegiate education is the cadre of general education courses required for all majors. All disciplines build upon this component and ours is no different [6]. Two of the most important aspects of general education that should be threaded throughout the coursework are problem solving and critical thinking. They should permeate all baccalaureate curricula. Professor Arnold Arons in his paper, "Critical Thinking and the Baccalaureate Curriculum", states a few of the thinking and reasoning processes that underlie analysis and inquiry involve consciously raising the questions, "What do we know ?", "How do we know ?", "Why do we accept or believe ...?", "What is the evidence for ...?". He further explains that the "How do we know ?" should be an intrinsic part of general education.

Isaksen and Parnes [10] emphasize that "effective problem solving is based upon both creative as well as critical types of thinking". They point out that creative thinking involves making and communicating meaningful connections, thinking of many possibilities and alternatives, becoming aware of problems, deficiencies, and gaps in knowledge, bringing together available information, making hypotheses and modifying and retesting them, and finally communicating results.

There is little disagreement among college faculty that students' thinking abilities must be improved; however, the discussion continues concerning the most effective means to attain this end. Some colleges teach critical thinking as a separate course while others incorporate it into general education and thus across all curricula; the latter being the method used at Kean. "Regardless of the route taken, critical-thinking experts say there is no foolproof way to teach thinking. What is certain, they say, is that faculty members must be ready to change the way they approach the learning process. Most thinking activities shun traditional teaching" [8]. However, writing assignments, in particular "freewriting", are very significant for teaching thinking since the student must clarify his/her thoughts before writing them.

Professor Robert Sternberg of Yale University explains that students need help in recognizing problems, not just in solving them [11]. He describes possible ways of teaching critical thinking and states "good thinking in one academic or particular area of endeavor does not guarantee good thinking in another". This is supported by a recent article examining the relationship between programming and thinking [13]. Professor Sternberg continues "critical thinking programs need to sample a variety of thinking skills and to sample them in ways that are true to the way problems appear in our everyday lives" [12].

The Application-Software-Critical-Thinking Connection

If we consider an application software course in which the computer is presented as a tool, it is clear that students will develop the lower level thinking skills of knowledge and understanding concerning how to use the specific software; namely, entering data, searching and sorting files, inserting and deleting text, etc. However, critical thinking as we have seen requires higher level thinking skills. We believe these can be developed by presenting application software, such as word processing, spreadsheets, and data base, in such a manner as to examine both textual and numeric information , formulate an analysis, and answer appropriate questions. Let us consider a few examples.

Using word processing facilitates freewriting and thus enables the student to develop his ideas with greater fluidity due to all the editing features. These features ease the rewrite process which, in turn, allows the student to fine tune his thoughts. Just as one becomes a better reader by reading (Duquesne University uses a specialized software package in teaching critical reading [14]), one also becomes more proficient at writing (and thinking) by writing. The facilities of word processing for reorganizing and synthesizing paragraphs to develop a topic should assist the student in fostering better writing skills. An instructor could easily present an unstructured essay and have the students use the word processing features of insert, delete, move, copy, etc. to create a well-developed expository one. Comparisons of text can be readily viewed. Outlines can be dynamically expanded within a document. The software becomes a pedagogical tool.

In addition, if the student has had some exposure to programming (many freshmen enter college with such an experience), the parallel between writing and programming should be explicitly noted for the student. They are similar inasmuch as both are creative, mental activities involving definition (topic in the former and problem statement in the latter), analysis, structure and syntax, stepwise refinement, and documentation.

If we wish to perform numeric analysis and formulation, then electronic spreadsheets provide a quick and easy means. In particular, consider a budget analysis using a spreadsheet. Every student usually has a working knowledge of a budget and realizes the implications inherent in one . Also, budget-type problems lend themselves to spreadsheet analysis. "Additional advantages to using spreadsheets include 'what if' experimentation. Students can alter values searching for the "best" solution to a problem." [5] They are able to examine the various parameters in a problem and to explore the connections between them as well as test a hypothesis concerning the relationship. The immediate feedback factor supports student exploration. "Try it and see what happens" becomes a natural expectation and initiates an analysis. Consequently, the students become involved with creative thinking.

This type of numeric analysis does not require a high degree of mathematical sophistication; high school algebra is sufficient. In addition, the student who has a history of "math anxiety" is free to explore.

When we have students working with data bases, they begin to examine the relationship between data and information. If we design the appropriate questions, the student will formulate a response that was generated by data base analysis. Hypothesis-testing is possible and discussion of what is relevant data should occur.

Consider a data base of the fifty states. The level of thinking skills vary depending upon the type of questions posed. To list all the state capitals is very low level and generating the ten most populated states is slightly above that. However, if we wish to know which ten states have had the greatest population shift in the last ten years, the level increases considerably. It does so even more when we look to the data base for some possible reasons for these population shifts. The student examines the data with a more discriminating eye. Questions arise as to what type of information can you get from a data base. Consideration is given to the design of the given data base and what would one need to obtain from the data base to test a hypothesis. All of this involves the higher-level thinking skills of analysis and evaluation. Dr. Janet Parker states very succinctly in her paper, "Tools for Thought" [15], "Getting data from a data base or spreadsheet is only the beginning; interpretations and inferences need to be drawn. Students need to be pushed beyond simple knowledge of the data, toward analysis, synthesis and evaluation. You will have to work to take students beyond the level of simple data input, recall and listing, toward evaluating what they see, making inferences about what it means, and coming up with some kind of meaning in terms of solving questions or problems."

The General Education Problem

We see several problems arising with a course that covers the content that we have described being considered as a general education offering.

First and possibly foremost, is this really a general education course that academicians can feel comfortable offering for the broad population of students? We believe it is and it manifests a twofold benefit. The faculty member has a new tool with which to approach the learning process and technology (although low level) is introduced to all students. The student gains elementary skills with this introduction, but more importantly begins to consider and analyze the information generated within a technological society.

Another difficulty lies in determining which faculty will teach this. Let us immediately state that it should not be the computer science faculty. This course is neither a service course nor a software survey course, but rather should be viewed as a component of general education. We support the statement that the use of applications packages should be considered a collegiate rather than departmental responsibility.[16] At Kean College the faculty have developed the following core of six GE courses:

Composition,
Inquiry and Research,
Emergence of the Modern World,
Intellectual and Cultural Traditions of Western
  Civilization: 1450 to the Present,
Landmarks of World Literature, and
Science and Technology in the Modern World.

These courses are taught by professors from varied disciplines and from different schools within the college. It is our belief that the proposed course should be handled in the same manner. However, concern may arise about the qualifications of an individual faculty member to teach this. We believe most faculty will not be prepared to teach this outright and that some training and workshops would be necessary, especially to insure that the course does not deteriorate to acquiring only low level skills.

We agree that computers and their software are not neutral tools and that they have the capacity to amplify as well as reduce different aspects of learning.[17] That is why this course cannot stand alone. It must be prefixed by or collateral with other courses that develop problem solving ability and analytical reasoning. (At Kean the thrust of the new GE core is just that.) Some may suggest that the content we describe could be incorporated within another course. We caution against that since we feel that the material will not be given adequate coverage from both the critical thinking and skill-based perspective.

Since the course would require "hands-on", students need ready access to equipment. This may not be a problem at some institutions, but would put a definite strain on those with limited resources. The entire issue of labs raises many concerns involving staffing, maintenance, availability of software, etc. These are problems that must be addressed college wide and require administrative decisions.

SUMMARY

Our experience in teaching information processing to management science and accounting majors via software productivity tools has led us to the belief that more can be accomplished. Processing information and using software to assist in problem solving is relevant to all students, not just a select few being serviced in a specific major. Thus we have set forth a proposal for a general education course that may raise many questions for which we have no ready answers. However, we believe it is worth considering. No attempt has been made to present an outline for such a course since the proposal itself warrants more discussion and research. (Dialogue has begun with the coordinator of the General Education Program.)

This proposal has been generated by the belief that the fabric of critical thinking is woven with analysis and synthesis of information and can be threaded with various application software. It is the hope of the authors that this paper will spark (potentially many), ignite (probably some), engulf (possibly a few), and not burn out.

REFERENCES

[1] Anderson,R., Chairman's Message, *ACM SIGCUE Bulletin*, Vol.17, No.2, Spring 1983, p.1.

[2] Abrahams,P., "What is Computer Science", President's Letter, *Communications of the ACM*, Vol.30, No.6, June 1987, pp.472-473.

[3] Halaris,A. and Sloan,L., "Towards a Definition of Computer Literacy for the Liberal Arts Environment", *ACM SIGCSE Bulletin*, Vol.17, No.1, March 1985,

pp.320-326.

[4] Dyck,V.,Black,J., and Fenton,S.,"Beyond Traditional Computer Literacy", *ACM SIGCSE Bulletin*, Vol.19, No.1, February 1987, pp.508-512.

[5] Bailey,M., "Spreadsheets and Databases - Alternatives to Programming for Non-Computer Science Majors", *ACM SIGCSE Bulletin*, Vol.19, No.1, February 1987, pp.499-503.

[6] Computing Sciences Accreditation Board, Inc., Criteria for Accrediting Programs in Computer Science in the United States, January 23, 1987.

[7] Arons,A.,"Critical Thinking and the Baccalaureate Curriculum", *Liberal Education*, Summer 1985, pp.141-157.

[8] McMillen,L., "Many Professors Now Start at the Beginning by Teaching Their Students How to Think", *Chronicle of Higher Education*, No.32, March 1986, pp.23-5.

[9] Stuckey,J., "The Computing Skills Workshop at Carnegie-Mellon University", *ACM SIGCUE Outlook*, Spring/Summer 1986, pp.46-48.

[10] Isaksen S. and Parnes,S., "Curriculum Planning for Creative Thinking and Problem Solving", *The Journal of Creative Behavior*, No.19, 1985, pp.1-29.

[11] Sternberg,R., "Teaching Critical Thinking, Part 1: Are We Making Critical Mistakes?", *PHI DELTA KAPPAN*, November 1985, pp.194-198.

[12] Sternberg,R., "Teaching Critical Thinking, Part 2: Possible Solutions", *PHI DELTA KAPPAN*, December 1985, pp.277-280.

[13] Mayer,R.E.,Dyck,J.L., and Vilberg,W.,"Learning To Program and Learning To Think: What's the Connection?", *Communications of the ACM*, Vol.29, No.7, July 1986, pp.605-610.

[14] Orndorff,J., "Computer Aids for the Teaching of Critical Reading in the Liberal Arts Curriculum", *Proceedings of National Educational Computer Conference* June 1987, pp.208-212.

[15] Parker,J., "Tools for Thought", Journal of the International Council for Computers in Education, *The Computing Teacher* Conference Issue, 1987, pp.11-12.

[16] Parker,J.D. and Schneider, G.M., "Problems With and Proposals for Service Courses in Computer Science", *ACM SIGCSE Bulletin*, Vol.19, No.1, February 1987, pp.423-427.

[17] Bowers,C.A., "Teaching a 19th Century Mode of Thinking Through a 20th Century Machine", *Proceedings of the National Educational Computer Conference*, June 1987, pp.114-118.

[18] Kean College of New Jersey, "CPS 1032 Course Outline: Principles of Information Processing Using Microcomputers", February 1987.

[19] Slavin,R.E., "Learning Together", *American Educator*, Vol.10, No.2, Summer 1986, pp.6-13.

[20] Levine,M., "Docemur Docendo (He Who Teaches, Learns)", *American Educator*,Vol.10, No.3, Fall 1986, pp.22-25,48.

[21] Spence,J. and Windsor,J., "Establishing and Managing a Microcomputer Laboratory - Part III", *Interface: The Computer Education Quarterly*,Vol.8, Issue 4, Winter 1986-87, pp.12-17.

CATALYSTS FOR CHANGE BASED ON TEACHERS' CONCERNS WITH MICROCOMPUTERS IN INSTRUCTION

Rachelle S. Heller and C. Dianne Martin
Electrical Engineering and Computer Science
The George Washington University

ABSTRACT

Whether an innovation, such as microcomputers in instruction, becomes permanently institutionalized into the curriculum depends in major part on the attitudes and concerns of the teachers. In this study 495 teachers from three large suburban school districts were polled regarding the use of microcomputers in instruction The measure, Stages of Concern Questionnaire (SoCQ) was developed by Hall et al (1979) and is based on seven stages of concern. The teachers in this study were all involved in their first year of microcomputer use in instruction as part of the official curriculum of their school districts. The resulting composite profiles show a high level of personal concern characterized as a nonuser profile, regardless of computer training, years of teaching experience, or level of use of the microcomputers. Reasons for this phenomenon and suggestions for ways to alleviate it are described.

INTRODUCTION

Educational settings are particularly resistant to change, and the classroom teacher has proven to be a key actor in the success of any innovation in instruction. This is particularly true for the innovation of microcomputers for instruction. The literature (Martin & Heller, 1985, Winkler, et al, 1986) suggests that while the computer is an anxiety-producing innovation for teachers due to its complex, demanding and non-obvious nature (Meister, 1984), exposure to computers through usage and training should alleviate much of the concern. If this is true, we would expect to find differences in the concerns of teachers who are long-time computer users and those who are not, and between teachers who have had formal computer training and those who have not.

There have been many studies conducted over the past fifty years about the general anxieties and concerns of teachers (Fuller, 1969). In 1974, a progression of seven stages of concern (SoC) was identified to describe concerns felt by users of an innovation as they become increasingly skillful in using an innovation (Hall et al., 1979). A concern is defined to be "the composite representation of the feelings, preoccupation, thought and consideration given to a particular issue or task" (Hall et al., 1979 p. 5). The SoC progression serves as a basic dimension for describing the dynamics of an individual innovation adopter, such as a classroom teacher, and is used in this study to describe teacher concerns over the innovation of microcomputers in instruction. The seven stages of concern, numbered from 0 to 6, identified by Hall et al. are:

0) *awareness* - the initial contact with an innovation in which there is little concern or involvement with the innovation.

1) *informational* - a general awareness and desire to learn about the innovation, but the individual does not see the relationship of the innovation to self.

2) *personal* - individual begins to worry about the impact and demands that the innovation will make upon his/her situation and about personal inadequacy in dealing with the innovation; financial and status considerations for self and colleagues may also be of concern.

3) *management* - attention is on the process and tasks involved in using the innovation in own environment; issues related to efficiency, organization, management, and time are most important.

4) *consequence* - attention is on the impact of the innovation on the learning environment and on the students within the sphere of influence of the teacher; focus is on motivation, student performance, and changes in the classroom caused by the innovation.

5) *collaboration* - attention is on coordination and cooperation with colleagues regarding the use of the innovation.

6) *refocusing* - attention is on exploring additional benefits and uses of the innovation as well as refinement of current use to increase effectiveness of the innovation; may include alternatives to the innovation that may appear to be more effective.

Martin and Heller (1985) describe the first three levels of concerns as *self-directed,* in that they focus on the effect of the innovation upon an individual, such as a teacher. The next two levels of concern are *task-directed* and focus on concern about the innovation as it is related to an environment, such as a classroom. The last two levels of concern were *impact-directed* and focus on the impact of the innovation across a set of environments, such as a school district. Individuals involved with an innovation will go through progressive stages of concern as they become more sophisticated users of the innovation.

METHODOLOGY

In this study a total of 495 teachers from three large suburban school districts were administered the Stages of Concern Questionnaire (SoCQ) to determine their level of concern regarding the use of microcomputers for instruction. These teachers had varied backgrounds. They had from one to 25 years of teaching experience. Some of them had received formal computer training and some of them had not. Most of them were regular classroom teachers from grades K to 8. A few of them were specialists, such as speech therapists, special education or resource teachers. The common denominator for all of these teachers was that they were involved in their first year of using microcomputers for instruction in an officially sanctioned program of their school district. The instrument used to collect this data was the 35-item SoCQ and the SoCQ Quick Scoring Device (Hall et al., 1979). At each administration of the SoCQ for this study the participants were told to substitute the words "computers in the classroom" or "educational uses of computers" for the word "innovation" to determine the level of concern related to computers in the classroom.

District A Characteristics

District A is comprised of a large county encompassing 500 square miles with a population mix of 59% white, 37% black, and 4% other minorities. This school district is a suburb of a major metropolitan area. According to the National Center for Educational Statistics, this county's school system would rank among the country's top 1% of school districts based on student population (102,000).

Seven hundred thirty-five questionnaires were mailed in the late spring of 1986 and 324 usable responses were received. This 44% response rate represents replies from teachers, aides, area specialists and administrators after they participated in one full school year of the the first year of instructional use of computers. The instructional program was officially implemented into the county's schools in the fall of 1985 and was based on a detailed, county-wide plan. The plan included a minimum of one 40-minute computer session per week for each student. All of the educators involved in this study received at least a one day inservice training course by the county specialists. Yearly on-going support was offered to all teachers by these same specialists.

District B Characteristics

District B coincides with a county local government unit encompassing 527 square miles of land area with a total population of approximately 70,000 people and a medium family income of $43,393 as of April, 1986. Until recently, the county had been mostly rural, but with a rapidly expanding population migrating from the neighboring county, it is quickly becoming suburban with the attendant new demands on its school system. The county has a predominantly white population, with less than 10% black and less than 5% other minority members combined. Located one county removed from a major metropolitan area, with the eastern side rapidly becoming a suburban extension of the metropolitan area, it is representative of school districts in the second ring of suburbanization around a major metropolitan area. The school district has an enrollment of 13,000 students and is in the upper 3% of school districts by size in the country.

Data were collected about the concerns of 114 teachers in the school district who participated in a three-credit graduate course, Utilizing Computers in the Classroom, taught in the fall 1984, spring 1985, fall 1985, and spring 1986, and subsidized by the school district. Instructional uses of computers were officially implemented into the county curriculum during the 1984-86 school years. All of the teachers in the study were from schools receiving computers and were involved in using them in their classes. The SoCQ was administered at the beginning and/or end of the course to determine the level of concern of the participants regarding the innovation before and/or after the computer course. The SoCQ was administered at the end of the computer course during two sections of the fall 1984 semester. The SoCQ was administered at the beginning of the computer course during one section of the spring 1985 and spring 1986 semesters. During the fall 1985 semester, the SoCQ was administered at the beginning and the end of the course. There were 40 pretests only, 50 posttests only, and one group of 24 with both pretests and posttests.

District C Characteristics

This small suburban county of only 25.8 square miles has more than 152,000 residents, according to the 1980 census. Of that number, approximately 83% are white, 9% are black, 5% are Asian and other minorities comprise more than 2% of that total. This densely-populated bedroom community for a major metropolitan area is surrounded by other suburban counties with much more land mass and fewer per capita residents. The 1987 medium family (two persons or more) income is over $47,000. The 1986 school enrollment figures are just over 14,400 students, putting it in the upper 3% of school districts by size in the country and within the third of six class sizes used by National Center for Educational Statistics to classify school districts by size.

When the teachers from this county were studied, an instructional program for computers emphasizing LOGO programming was just being developed. The county committed itself to an inservice education program for its teachers and agreed to subsidize a three-credit, 14-week graduate course in computer literacy and LOGO programming. The 48 respondents to the SoCQ completed the course in fall 1983 or spring 1984. Very few of the teachers had computers in their classroom at the time they completed the SoCQ.

RESULTS AND DISCUSSION

Each individual has a unique concerns profile measured by the SoCQ. In this study the individual profiles each school district are averaged together to form the aggregate concerns profile for the districts shown in figures 1, 2, and 3. There is a striking similarity between the three aggregate profiles. In each case the self-directed concerns are higher that the task and impact-directed concerns. Further, within the self-directed concerns, the informational concern level is at least as high as the awareness concern level. Hall et al., (1979) describe this profile to be representative of a non-user (see Figure 4):

> In all of the research that has been done to date using the SoCQ, the nonuser concerns profile stands out most clearly and consistently. Nonusers' concerns are normally highest on Stages 0, 1, and 2 and lowest on Stages 4, 5, and 6. There is some variation in the amount of intensity of these concerns...however, the general shape of the pattern ...reflects the interested, not...over-concerned, positively disposed nonuser (Hall et al., 1979, p. 36).

A nonuser is *not* someone who is not using the innovation at all. Rather, a nonuser is someone whose personal concerns about the innovation outweigh their concerns about implementing the innovation into their environment. The authors suggest that this might impede the implementation of the innovation. The teachers in the study were, in fact, using the computers, turning them on and off, inserting diskettes, and even assisting children in the use of the computer. Because their self-directed concerns are significantly higher then their task-directed concerns, they are still considered non-users. Even when factors such as years of teaching experience, amount of computer training, or present level of use of the microcomputer in instruction are used to produce concerns profiles for subgroups of teachers in District A, the resulting profiles (Figures 5a, 5b, and 5c) are still those of nonusers and are not significantly different than the overall composite profiles.

The data reveal that all first year, computer-using teachers exhibit the high self-directed concerns of the nonuser. The teachers in District C who were involved only in a computer course during this first year exhibit the nonuser profile. The teachers in District A who were involved with in-class use of computers exhibit the nonuser profile. The teachers in District B who were both receiving training and using computers in class also exhibit the nonuser profile.

CONCLUSIONS

This phenomenon can be explained by the interaction of an individual teacher with the organizational structure of a school district and with the characteristics of the innovation of microcomputers (Martin, 1987). Teachers tend to be isolated from each other and from the other operational levels of the school district. Unlike administrators, they usually do not have access to the big picture about the direction that the educational use of computers will take in the school district. Not only do teachers lack a comprehensive view of computers in education, they are not well informed about the theory and practice of using computers within their own classrooms, in part because most computer literacy programs for teachers have focused on the operation of computers and not the educational theories and values derived from their use. In addition, computers are viewed as a highly technical and complex innovation by teachers (Rogers, 1985). Because the computer is a multi-purpose device and can be put to many uses, the teacher is easily overwhelmed by the computer. In many cases teachers have expressed fears about damaging the computer or the software.

A second feature exhibited by all of the profiles is a low level of task-directed concern. This indicates that the teachers are not yet focused on the management of the computer in the educational process or concerned about the consequences of using computers in the classroom with students. Previous research using the SoCQ suggests that high self-directed concerns will take priority over the task-oriented concerns.

Some of the composite profiles (Figures 2 and 3) display a high level of collaborative concern. These profiles are from groups of teachers who were trained together or who work closely together. The fact that this phenomenon is not widespread is related to the fact that most teachers work in isolation from each other in a school district, and without some formally-established opportunity for a shared experience, there are no convenient mechanisms set up for collaboration with colleagues.

The full potential of microcomputers in instruction will not be realized until the concerns of teachers move beyond the self-directed, personal level. Generally, a person with a nonuser profile will not successfully accept or utilize the innovation until his or her personal concerns are lowered. Many might argue that their self-directed concerns might actually impede innovation. Although many of the profiles show an upturn at the impact-directed level of concerns, the task-directed levels of management and consequence concerns are relatively low on all profiles. The task-directed concerns are critical because they provide impetus for the actual integration of computers in the instructional process by teachers in the classrooms.

Time will serve to help teachers lower their self-directed concerns. While this passive approach has its merits, there are some actions that can be taken to create a positive environment in which teachers can address their concerns. As the profiles indicate (Figure 5b), just the passage of time has not been sufficient to move the teachers from a non-user profile. As has been pointed out by Hall et al., "Higher level of concerns cannot be engineered by an outside agent" (p.6). Change must come from within the individual; however, an atmosphere conducive to change can be created by outside catalysts.

The first change catalyst must address the isolation and anxiety felt by many teachers. Fostering communication networks among teachers is crucial to alleviating this stress. Such communication networks can take the form of scheduled meetings, newsletters, or even electronic mail systems. The focus of this networking should be on sharing experiences, so that the personal feelings, concerns and even fears of the individual teachers can be addressed and alleviated. The sharing of both successes and failures should be encouraged to legitimize teachers' self-directed concerns and help them to move to higher levels of concern.

The second catalyst is appropriate teacher training. Many teachers leave their first computer course with higher anxieties about computers than when they entered (Figure 3). Since the first computer course for teachers primarily addresses the information level of concern (Martin & Heller, 1985), it is clear that more training is needed by teachers after they have had a chance to use the computer with their students. Many school districts are finding now that the demand for further training by teachers who have had one course is as great as the initial demand for training was five years ago.

Previous attempts to deal with task-directed concerns in a first computer course for teachers have failed because teachers were simply not ready for them. As Hall suggests, impetus can't be a push from the outside. "Providing inputs that are not stage relevant (e.g., attempting to force high level concerns) is an assured way to increase the intensity of lower stage concerns. Whether and with what speed higher level concerns develop will depend upon the person as well as the innovation and the environmental context " (Hall et al., 1979, p. 6). It is in a second-level computer training course that the task-directed concerns and issues can be effectively addressed. It is also important that the teachers themselves help to shape the content of the second-level courses (McLaughlin &

Berman, 1977) and that the courses be designed to deal specifically with teacher concerns as well as the technical, practical, and theoretical issues of teaching with computers.

A third catalyst for creating an environment to encourage teachers to move beyond self-directed concerns is with a good technical support system that is non-judgmental and easily accessible. "The provision of adequate technical support is an important way of demonstrating administrative commitment to an innovation" (Winkler, Stasz, & Shavelson, 1986, p. 23). Such technical support includes a sufficient amount of hardware and software for instructional uses and sufficient technical staff to troubleshoot problems over the phone and on-site when necessary. Each school should have a staff person available with enough expertise to handle low level technical questions, thereby allowing for quick answers to questions, and freeing the technical support staff for the more difficult problems.

A fourth catalyst suggested by Winkler et al. is to provide incentives such as release time, pay bonuses, and opportunities for career advancement based upon the teacher's willingness to upgrade skills. The use of teachers to create new curriculum and material and to conduct training for other teachers has also been cited as an incentive since it provides recognition from peers for expertise.

Ultimately, the self-directed concerns of teachers are related to the isolation, the time constraints and the lack of professionalization that are inherent with their job. In this study we have identified the level of teacher concerns related to the use of microcomputers in instruction and have offered some short term solutions. The significance of these solutions is that they emerged from the current levels of concern of teachers about microcomputers, rather than addressing concerns not felt by teachers. In the long run, however, the best solution is the increased professionalization of the job of teaching in which teachers are provided with release time, incentives to innovate, and opportunities to collaborate with colleagues on a regular basis.

ACKNOWLEDGMENT

The authors wish to thank University of Maryland professors Dr. Mark Gillingham and Dr. Gary Marchionini for their contribution to the collection and analysis of data from District A.

REFERENCES

Fuller, F.F. "Concerns of Teachers: A Developmental Conceptualization." *American Educational Research Journal,* Vol. 6, 1969, pp. 207-397.

Hall, G. E., George, A. A., & Rutherford, W. L. Measuring Stages of Concern about the Innovation: A Manual for Use of the SoC Questionnaire. Research and Development Center for Teacher Education, University of Texas at Austin, 1979.

Martin, C. D. School District Implementation of Microcomputers for Instruction. Doctoral dissertation, George Washington University, Washington, D. C., 1987

Martin, C. D. & Heller, R. S. "Analyzing Teacher Training in LOGO Using a Stages-of- Concern Taxonomy." *Proceedings of LOGO '85*. MIT Press, Cambridge, MA., 1985.

Meister, G. R. *Successful Integration of Microcomputers in an Elementary School.* California: Stanford University, School of Education, Institute for Research on Educational Finance and Governance, 1984.

McLaughlin, M. W. & Berman P. "Retooling Staff Development in a Period of Retrenchment ." *Educational Leadership*, December, 1977, pp. 191-194.

Rogers, E. M et al. Microcomputers in the Schools: A Case of Decentralized Diffusion. Stanford University Institute for Communication Research, CA, 1985.

Winkler, J. D., Stasz, C., & Shavelson, R. Administrative Policies of Increasing the Use of Microcomputers in Instruction (Rand Report R-3409-NIE). Rand Corporation, Santa Monica, CA, 1986.

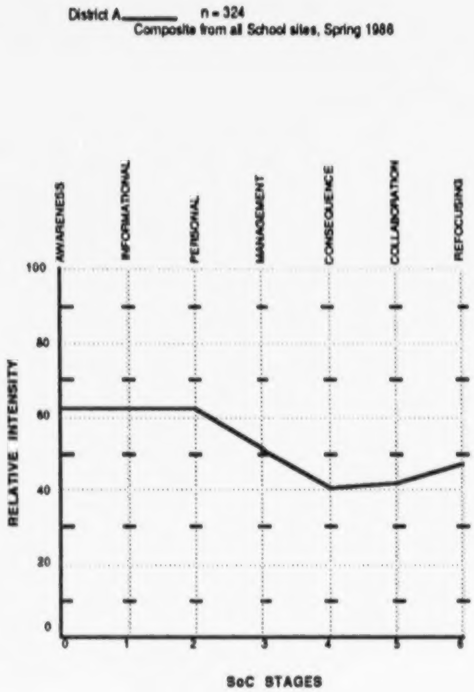


Figure 1

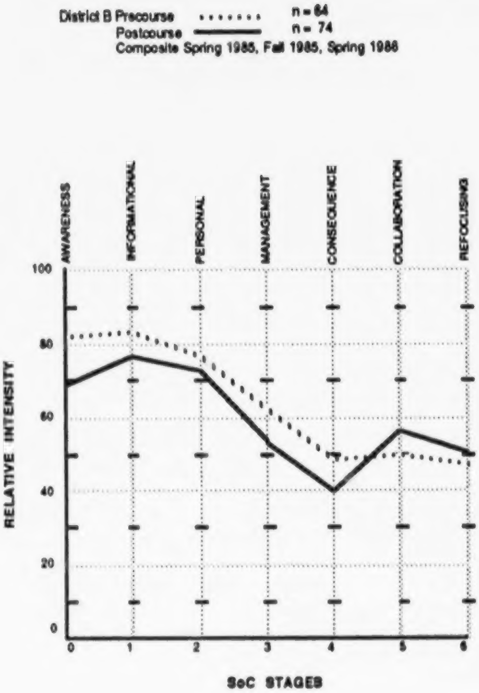


Figure 2

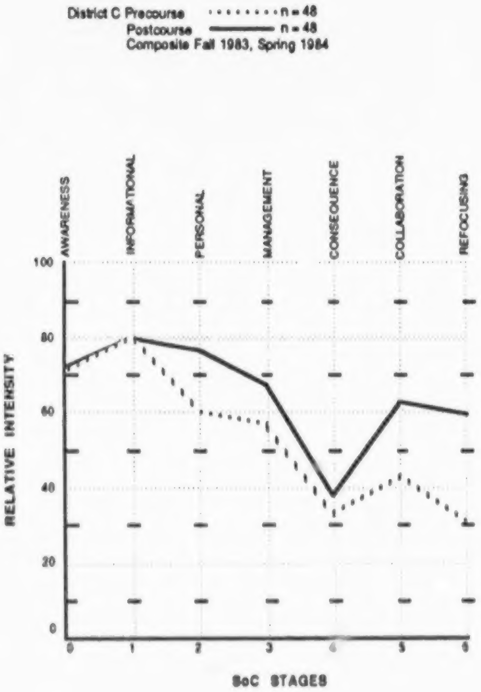


Figure 3

Typical Nonuser SocQ Profile

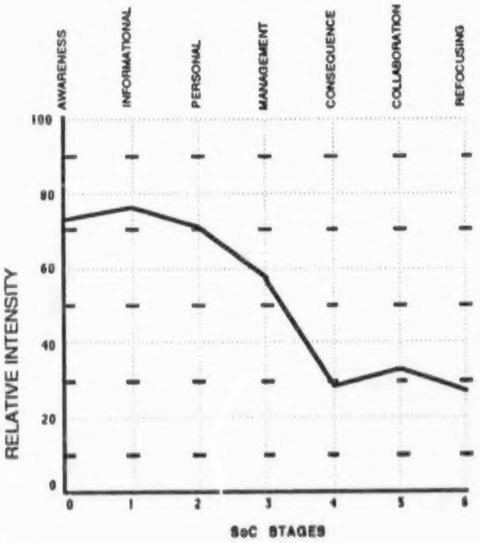


Figure 4

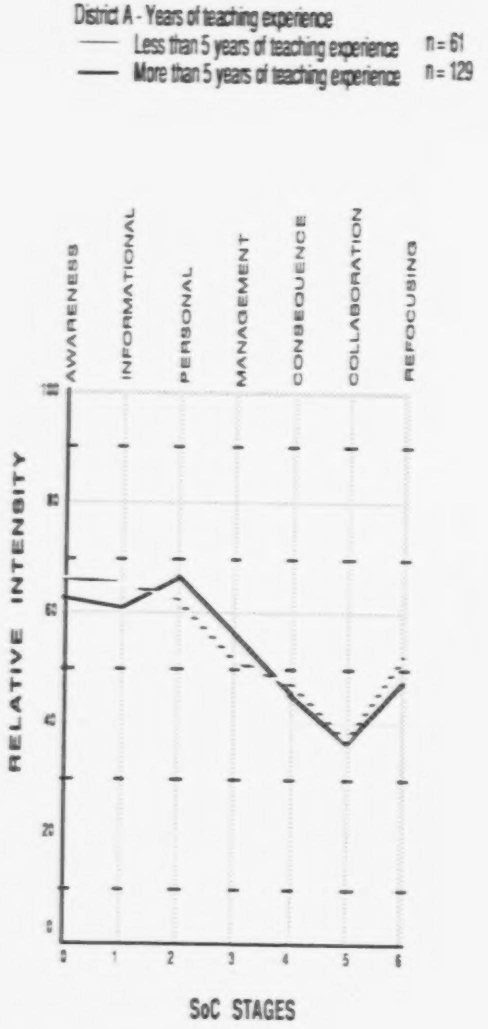


Figure 5a

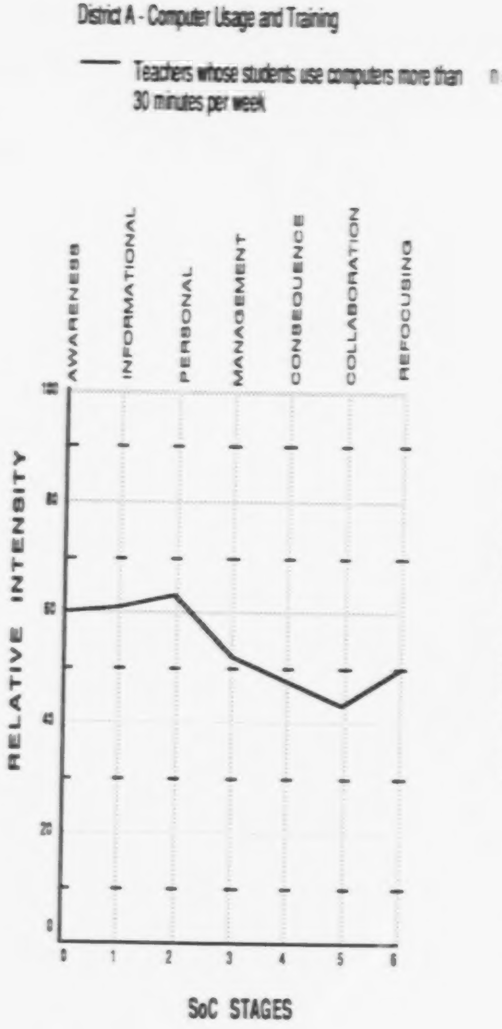


Figure 5b

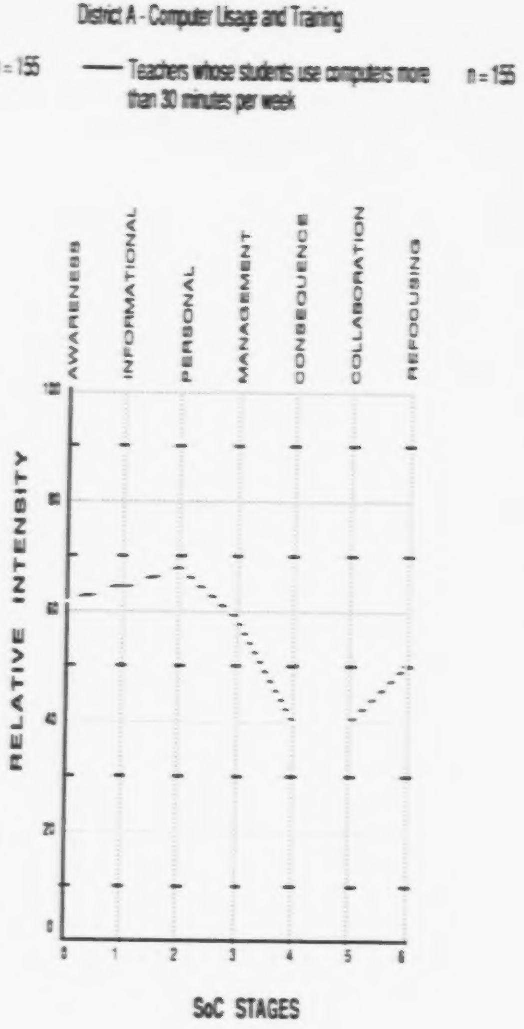


Figure 5c

TEACHERS AS COLLABORATIVE RESEARCHERS: PROFESSIONAL DEVELOPMENT THROUGH ASSESSING LOGO LEARNING

Molly L. Watt and Daniel H. Watt,
Senior Associates
Education Development Center
Center for Learning Technology
55 Chapel Street
Newton Massachusetts 02160
(617) 969-7100

ABSTRACT

This paper describes a series of collaborative research seminars designed to allow classroom teachers to observe and assess critical aspects of Logo learning in their own classrooms. We define critical aspects of Logo to be those concepts drawn from the domains of computer science, mathematical thinking and problem-solving which are essential for students to use and understand if they are to gain maximum benefit from their Logo learning experiences. This paper describes the approaches we developed to support classroom teachers in observing their own students, collecting data about their work, and assessing their learning of these aspects of Logo. We also describe our approach to assessing the professional development of teachers who participated with us as collaborative researchers.

This work was supported by the National Science Foundation as part of **Exploratory Research on Critical Aspects of Logo Learning**, grant number MDR 865 1600. We would also like to acknowledge the contributions of research consultants Roy Pea of New York University and Jan Hawkins of Bank Street College of Education, our Research Assistant, Joan Funk and Administrative Assistant, Gerry Sills of Education Development Center, and collaborating teacher researchers, Elizabeth Berrner, David Crump, Diana Freedman, Doreen Kelly, Robert Kondel, Jane Manzelli, Mary Miller-Teehan, Pasquale Puleo and Kitsy Rothermel. Any ideas, opinions, findings and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation or of any of our collaborators.

I. INTRODUCTION AND BACKGROUND

Our project started with questions by teachers and findings from research which indicate that the educational potential of Logo is not yet fully realized in classrooms. We are often told, "Logo isn't working," or "Students aren't learning important mathematics and computer science through exploration and discovery." (See, for example: Pea, et. al. 1987; Kurland et. al. 1985; Watt, D. 1982; Moursund, 1983; Leron, 1985a,1985b). These opinions, distressing as they sound, match some of our own observations while teaching Logo to both children and teachers. The learning challenges our own students encounter demonstrate the types of difficulties experienced by many Logo learners.

As we compared these observations about student learning with the powerful ideas embedded into the Logo language itself by its developers (Papert, 1980; Watt D. 1982; Watt M. 1982; Watt and Watt, 1986), we began to develop the concept of critical aspects of Logo learning, aspects of Logo which when understood and used by a Logo programmer allow that person to use Logo powerfully. And in our experience, the most powerful uses of Logo involve project-oriented work of the learner's own choosing which leads to the learning of important mathematical and computer science ideas while in the process of carrying out the project. Critical aspects, therefore, can serve as a framework by which a teacher can assess and guide students as they work on their own projects.

For the past year we have been working with teachers as collaborative researchers to use the concept of critical aspects of Logo to identify and describe important Logo learning in their own classrooms. It was our expectation that teachers who participated with us in this work would be able to document and demonstrate the value of the Logo learning in their own classrooms. To support this process we conducted an ongoing series of seminars to involve the teachers in a community of colleagues with a shared purpose.

Part of our model for collaborative research seminars came from our own experiences as teachers and researchers. Our extensive work with the development of teacher centers encouraged us to value teachers' professionalism. We modeled some of our seminar structure on the Children's Thinking (Research) Seminars developed and led by Bill Hull and Sara Hull (Hull, 1979; Watt. M, 1979). Some of our ideas about supporting teacher investigations were learned from Evans, Bamberger and Duckworth (Evans, Stubbs, Duckworth, and Davis, 1981; Bamberger, Duckworth, and Lampert, 1981; Duckworth, 1986). In our research and development work with Patricia Carini at The Prospect Archive of Children's Work and Research Center, North Bennington, Vermont, we participated in developing procedures for teachers to support each others' classroom action research projects (Carini, 1979). Our participation in The North Dakota Study Group on Evaluation gives us many models of qualitative research, observation and documentation processes teachers can carry out in their own classrooms.

This collaborative research project embraces many of our concerns for educational reform. We desire to support teachers in developing a growing sense of their own professionalism. And we desire to support both teachers and students in using Logo in the powerful ways and for the powerful purposes envisioned by its developers (Papert, 1980). In this paper we focus on the processes and experiences of the teacher research seminars and on the professional development of participating teachers. Issues related to critical aspects of Logo, and student learning will be discussed elsewhere.

II. THE RESEARCH SEMINARS

Our design for a teacher research seminar was based on prior experience with collaborative research, and on a broad survey of research on professional development of practicing teachers. Of particular importance to us is the research on Stages of Concern, and the Concerns-Based Adoption Model (Hall, et. al, 1973; Loucks and Hall, 1979). The central finding of this work is that teachers engaged in implementing a classroom innovation go through a predictable development in their concerns about and attitudes towards that innovation. Teacher education efforts that take these concerns into account, provide opportunities for teachers to express their concerns, and adapt the content of training sessions, workshops or seminars to fit the concerns of the teachers involved, have a much better chance of changing educational practice than activities which are not designed in this way.

Other research findings which informed our seminar design showed that teachers are more likely to use new content and teaching approaches in their classroom if they see specific instructional approaches modeled in training sessions; have time to practice the approaches modeled, and adapt the materials to their own situations; learn substantially more content than they plan to teach, and engage in learning experiences at their own level, while preparing to use a new approach with children; have opportunities to reflect on their own experiences, talk things over with colleagues, and experience support over time; and are recognized for their own areas of expertise.

We began our seminar series with a group of twelve teacher/researchers: a team of two teachers and a computer coordinator from each of four Boston area cities and towns. We met for eleven sessions, three all-day meetings and eight after-school meetings from 4:00 - 6:30 PM. Lunches were provided for the all-day sessions, and the teachers and project staff took turns contributing refreshments for the after-school meetings.

Each session began with a review and revision of the daily agenda, and ended with a four-question evaluation form. Each session was recorded, and parts of the recordings were transcribed for inclusion in the Notes and Commentary, compiled for each session by the project staff.

Activities varied from session to session. For example, the agenda for Session #3 (February 25, 1987) shows that the group started with refreshments; broke into triads to discuss "What do we look for when we observe students learning with Logo?"; reported back to the larger group; then broke into school-district teams to choose the two particular students whose work each teacher would follow for the rest of the year. The agenda for Session #6 (April 8, 1987) shows that the group started with refreshments; broke into two research groups focussing on examples of student Logo work brought in by teachers, each with a chairperson and a presenter; went on to share details about conference participation and presentation planning; and ended with a mathematical challenge for the researchers to solve in Logo.

The research seminars evolved through four distinct phases over the course of eleven sessions:

1. Learning to see: The first few weeks of the project were devoted to finding out what we were looking for and developing a common language to describe it.
2. Giving more structure to the research: These sessions were devoted to identifying and refining specific research questions, choosing students to follow, looking at the kinds of data available, and structuring the data collection and observations.
3. Looking at the patterns in the data: These sessions were devoted to attempting to understand what we were learning, and reflecting on our process of gathering data.
4. Writing project reports: At the end of the seminar series, the teachers wrote individual reports on their classroom research. Throughout the project, staff members wrote ongoing Notes and Commentary and progress reports in preparation for writing research reports.

We maintained a circulating library of shared resources: research reports, district Logo curriculum guides, Logo activity books and related magazines for participants to borrow. Most participants took advantage of this. Two members borrowed Logo research reports and most borrowed materials and books with Logo activities. We felt constrained by time limitations and did not get an annotated bibliography written for this group. We plan to continue the practice of sharing resources and expect to find more ways of putting relevant materials into each person's hands in future seminars.

The process we developed requires a group of committed participants, and several designated roles: a session leader; a note-taker; a Notes and Commentary writer; someone to copy disks and data brought in by participants for the group; someone to record the sessions; a librarian and food providers. These roles can be rotated or reassigned periodically.

We consider the teacher research seminar to be a generic form. The specific materials and approaches we developed, in this case the critical aspects of Logo, could be replaced by a different research focus.

III. OBSERVATION AND DATA COLLECTION TECHNIQUES

1. Structured Interviews

Each of the teacher/researchers in our seminar, besides being an experienced classroom teacher, had a minimum of two years experience teaching with Logo. Before beginning the seminar process, we conducted structured interviews with participants at their school sites. The interview gave us an opportunity to visit each classroom, meet with school administrators and understand the contexts in which our colleagues were working. It also provided a key source of data for understanding any professional development that might occur during the course of the seminars (see Section IV below). Finally, it allowed us to understand how the seminar participants thought about Logo, how they taught it, what their goals and objectives were, what they thought their students were learning, and how they knew whether their students were learning it.

The interview we used was designed by research consultants Roy Pea and Jan Hawkins, along with the project staff, to provide background information about each teacher's professional background, Logo experience, educational ideas and values, classroom situation, and Logo teaching space. It was designed to evoke a description about their Logo teaching practice by asking, for example: Describe your Logo curriculum. What content do you want your students to learn this year (cognitive objectives)? What is it important for your students to be able to do (performance objectives)? What other types of behavior or attitude changes do you hope for among your students as a result of Logo (Social objectives)? The interview went on to ask about teaching methods, where Logo fit into the day's schedule, the amount of Logo time spent both on and off computer and how else Logo work was structured. The section on teaching practices ended with questions asking about the teaching results: what students were learning, what discrepancies or surprises they had noticed between what they expected to happen and what actually had happened, differences among learners and any specific problems at specific stages of Logo learning.

The second section of the interview schedule asked about the Professional Development of the teacher/researcher. We asked how the participant began to use Logo, what workshops or courses had been taken, how they assessed their own knowledge of Logo, and what goals they had for their own continued learning of Logo. We asked how each teacher reflected on practice: how they set and monitored personal goals for teaching, and how they recorded plans, projects, ideas and problem areas. We also asked, "How do you improve and debug your own teaching?" We asked about their individual beliefs about Logo's importance and benefits for their students, how children learn it best and how they assess and monitor student knowledge and progress.

The third section asked questions about accountability and reporting practices with regard to their Logo teaching. The fourth section asked about their past experience with research and their expectations, hopes and dreams for this exploratory research project.

2. Using Printouts of Student Logo Programs as Data

The fifth section of the interview was almost an afterthought. We added it to the interview the day the principal investigators started the classroom visits and teacher interviews. At the end of each interview we asked each teacher to give us a printout of a Logo program by one student and tell us something about the program and why they chose to show it to us. We intended to use the printout to start the process of documenting and assessing Logo learning. We had assumed that printing out examples of student procedures and pictures would be part of a Logo teacher's usual practice. We were surprised that nine of our twelve collaborative researchers required technical assistance to print examples of student work. None of the researchers used printouts regularly to build archives of student work, or to assess learning in order to plan teaching interventions. Nor did they use them as models for their students to study. In fact several of the teachers reported that they rarely looked at a student's programming code at all unless a student asked them for

After we demonstrated how to print Logo procedures and graphics, we continued to work with teachers to define an appropriate form for presenting examples of student work to make them understandable for analysis of student learning. We developed a template indicating where to put the identification labels, dates and margins and a printout of the screen (see Figure 1). The procedures are printed out in order, with a procedure tree mapping the order in which each procedure and subprocedure is called. Then we helped participants learn how to do each of these steps. Later in the project, we developed some Logo tool procedures which generate procedure trees automatically and printout procedures in the correct calling order. (These Logo tools were adapted from models created by Richard Carter of Lesley College.)

3. What can printouts of students' procedures tell us about what they know?

We used student projects as data for a group process in assessing student work. Our goal was to describe the Logo, mathematics and computer science ideas that the student used in a particular project. Eventually we evolved Four Questions for looking at student examples. First we asked, "What does this student know?" Then, "What is not known by this student?" Next, "What can you suggest to this student to solve his problem - in his way?" And finally, "Based on what this student has already done, what might he be ready to learn next?"

Sometimes we looked at copies of a student's printout and ran the computer program. Some times we actually rearranged and added code to the program in an attempt to improve our understanding about how the student was thinking about a project, and to develop strategies for coaching that student on debugging or programming difficulties.

This process proved to be extremely fruitful for assessing the Logo knowledge a student was using. It often became the basis for diagnostic teaching using an emerging curriculum. When we studied several printouts of examples from the same student produced over a span of time, it was fairly straightforward to document increased Logo, mathematical and computer science knowledge. With practice, teachers could identify development and describe it in specific terms. For example, see Figure 2, showing two examples of the work of Kathy, a fourth grader, collected several weeks apart.

4. Critical Aspects and Teacher Interviews

We used our (evolving) critical aspects checklist as a reference for studying printouts of student work (see Figure 3). To use the list effectively, we found it necessary to focus on one or two clusters of critical aspects at a time. The whole list was just too large to use as a clear focus for observation.

We experimented, and eventually found a way to use the critical aspects checklist as a structure for an interview. This gave teacher/researchers another way to access their own knowledge about what a particular student knew about Logo. (But we had to practice the process several times before we learned to use it effectively -- one teacher/researcher described a seminar demonstration of a preliminary attempt as "deadly"!) An example of the way we might use the interview is to ask the teacher what the student knows about using procedures.

By looking at student examples we saw that that Sally and Pattie, whose keystrokes were stored in dribble files, probably did not know how to define procedures (see Figure 4). Kathy's show she was learning to combine steps and eventually define her projects in smaller conceptual pieces (see Figure 2). Whereas Heather shows us in her diagrams a fairly sophisticated way of using a superprocedure as a planning device (see Figure 5).

We developed and used this interview form towards the end of our project, and anticipate that teacher/researchers will be able to use it to support each other in future seminars. We expect that this process can also be used by experienced teachers, working with colleagues in their own schools, to structure their assessments of student knowledge.

5. Group Discussions and Notes and Commentary

In our research meetings we used discussions and group conversations as another way to support teacher/researchers in gaining access to their own knowledge and observations. We audiotaped most group discussions as part of the project documentation, and these tapes were used to write a set of Notes and Commentary on each meeting. The writer of Notes and Commentary reflected on some of the key ideas, insights and concerns raised in discussions and often quoted a participant's exact words or program examples (Watt, M. 1979; Carini, 1979; and Schon, 1983). Participants reading the Notes and Commentary found that it provided another opportunity to reflect on their experiences and to synthesize their own knowledge.

We also used the tapes of one meeting to help set the agenda for the next one. In this way we modeled some diagnostic teaching strategies within the research seminars themselves, and were able to work closely with the participants to support individual and group development and concerns.

6. Seminar evaluation Questions

Another form of seminar assessment data came from the responses to the four evaluation questions we asked each person to answer at the end of every session:

1. Question #1: What was the most important learning insight you had at today's meeting?
2. Question #2: What were you most perplexed by or concerned about at today's meeting?
3. Question #3: What would you like to talk about - or do - in future research meetings?
4. Question #4: What help, collaboration, or resources would support your continued work?

Answers to these questions were collated into a group response sheet and copied for everyone.

7. A case-study process to support research

Another specific technique we developed is a case-study process to facilitate a teacher/researcher in presenting data and research questions about a student's Logo work. Its design is based on previous work by Molly Watt (and others) while she was a Program Associate at the Prospect Archive and Research Center, North Bennington, Vermont, with Patricia Carini. The process was formalized, with specific roles and procedures, so that it could be used as a model, first for teacher/researchers in the seminar, and later for teachers to use with students in the classroom (Carini, 1979 and Martin, 1987).

Here's how this case-study approach works: Suppose a teacher has brought in several examples of a student's work to assess and wonders about the logic in the way the student is naming procedures. The procedure names seem miscellaneous to her and it is the teacher's impression that this student is often confused about how to find his project because he can't remember its most recent name or the names of its component parts. She wants the group to look with her at the student's work, trying to make sense of whatever logic she can identify and then have several strategies in mind for supporting the student in finding a scheme which would serve the student better.

These concerns lead to a focussing question which the teacher gives to the chairperson. (If the teacher does not have a clear focussing question it is the chairperson's responsibility to support her in finding one before meeting with the group.) The chair examines the data and makes sure that there is a computer for demonstration, if needed, or that printouts or journal notes are reproduced for the group to easily see. She then calls the meeting together, introduces the teacher, any description needed, the focusing question, the data and any constraints.

The teacher/presenter then describes the data, as specifically as possible, avoiding judgmental statements, and allowing the meaning to emerge from the description, the process and the data. The chair may invite the group to add descriptions of what they notice, without seeking consensus or discussing interpretations. The chair facilitates, takes notes and periodically summarizes. Then the chair asks for responses or suggestions about the focussing question. The group members offer specific insights. The chair writes down each suggestion without discussion or evaluation or questions. The presenter listens without responding to or evaluating the ideas offered.

After the group has completed supporting the presenter's observations, and responded to her focussing question, the members have a brief discussion of what was interesting and what they learned from the process about their own work and students.

Later it is the chair's task to write up notes about the meeting for everyone.

The development of this process was begun during our research seminars, and completed during the Assessing Logo Learning mini-course taught by Molly Watt and Stephen Shuller at The Logo Institute, August, 1987. In future research seminars, notes on research presentations will be integrated into the seminar process.

8. Individual Classroom Research Projects

The teacher/researchers focussed their classroom research around their own pedagogical questions. Some examples are:

> - How can I develop creative challenges to help students learn to use variables in Logo projects, which fall within my mathematics curriculum and the abilities of my students?
>
> - How can we get more students to use meaningful names for their subprocedures?
>
> - How can I get students to use superprocedures in a top-down programming style?

Questions about teaching strategies led to specific research questions, that could be used to determine the types of data that each teacher wanted to collect about what their students were learning. The teachers reformulated these questions several times, trying to find the right match between what they wanted to find out, and the data they were able to collect naturally as part of their on-going work.

Another way that teachers focussed their research was by choosing two students to observe, closely documenting their Logo learning. We asked them to choose one student about whom there was some particular aspect that interested them professionally. Then they chose a second student of the other sex, and with a different level of Logo expertise. (Our general categories were "struggling," "average," and "facile" Logo user.)

IV. PROFESSIONAL DEVELOPMENT

An important outcome of our research involves documenting the professional development of the teachers who participated with us as collaborative researchers. We have articulated a set of criteria for determining whether professional development occurred among participants in our research seminars:

- Do teachers indicate better understanding and articulation of what their students are learning?
- Are there any indications of changes in classroom practice during the process of this research project?
- Are there indications of willingness to share work with other professionals through writing and conference presentations?

The data we have collected, as a basis for analysis of professional development, include:

- pre/post structured interviews, described above
- pre/post Stages of Concern Questionnaires
- session-by-session materials: teachers' journal entries; teachers' examples of student work; teacher's research questions - original and revised; participants' evaluation forms for each session
- written research reports by teachers
- tape recordings of oral reports by teachers
- letters from participants about the project
- project information forms indicating future plans of participants
- information about presentations by participants at conferences and within their own districts

Preliminary review of our data with regard to the above criteria gives us reason to believe that a full analysis will demonstrate significant professional development on the part of participating teachers. We expect to complete the analysis by March 1988. The two illustrations that follow show what we consider to be indications of professional development.

The first example from our structured interviews with David Crump, a fourth grade teacher, demonstrates a change in his articulation of his approach to assessing student learning:

February Interview:

Q 25: What is your belief about how to assess what students know and monitor their progress?

A: My own experience in that area is based upon what students show me, and explain what they do. I have to say that I do get insight into how children are thinking when they ask me questions about how to do certain things, or to see that they at least know that they need to know something more to do what they want to do.

June Interview:

Q. 25: What is your belief about how to assess what students know and monitor their progress?

A: Over time. Seeing changes. And documenting those changes. And actually holding it up to some measure, whether that be the critical aspects or some other criteria. Seeing the physical evidence that students have been able to do some things. They've made some progress.

Q: In order to do this, you need to keep track of that information, because otherwise you can't compare it.

A: Yes. And it doesn't have to be every project or every time. It can be at the beginning of the year, in each of the next two quarters, and at the end of the year.

The second example from the same teacher is an excerpt from his final research report and indicates a change in his understanding about Logo's importance in the elementary school curriculum as a vehicle for students learning about mathematics, problem solving, programming and their own thinking.

Excerpt from "A Summary of the Classroom Research Conducted During the Logo Collaborative Research Project" by David Crump, June, 1987:

> *Finally I will reflect upon some differences in my way of thinking about Logo that resulted from my participation in the LCRP. Over the past five years I have been involved in either learning about or teaching Logo. I honestly felt that I knew all I ever needed to know about Logo in order to very successfully teach Logo in my classroom as delineated by our school curriculum.*
>
> *I recently became one of those educators who began to question why so much emphasis was being placed on the teaching and learning of Logo in elementary schools. I wondered if the benefits derived from children using Logo justified the time, effort and money invested in having it taught. Questions also surfaced in my mind concerning my own level of understanding of and knowledge about Logo. I began to feel that if I was more aware of some of*

the finer points of Logo, and understood better some of Logo's more powerful features, perhaps I might begin to see things in Logo that I'd been missing. When the opportunity became available to me to participate in the LCRP, I jumped at the chance, hoping to resolve some of the issues about Logo that concerned me.

The intensive training in the critical aspects of Logo learning that was a component of the LCRP seminars by far outweighs and outdistances anything I learned about Logo in workshops or courses. The training seminars constituted not only a redefining of what is important about Logo, but also a refining of Logo learning that I believe will enable me to be a much more effective teacher of Logo.

What I learned about recognizing and understanding the critical aspects of Logo learning, coupled with teaching strategies, methods of observation, as well as other material developed from this project related to methods of evaluating and enhancing Logo learning of students, is of greater value to me, and should be a model made available for other teachers of Logo. I believe that knowledge of and training in the critical aspects of Logo learning can provide a framework for curriculum developers who may need to reformat their present Logo curriculum, and develop new activities that provide the proper Logo experiences for students, so that they do progress in their learning and understanding of Logo.

I am now ready to justify in my own mind why the teaching of Logo is important. If taught the right way, students learn to think both logically and creatively. Not only do they have the opportunity to learn about Math, problem solving, and programming, but more importantly they begin to learn about their own thinking.

V. BIBLIOGRAPHY

Bamberger, J., Duckworth, E., and Lampert, M. (1981). *An Experiment in Teacher Development*, Final Report, NIE Grant # G-78-0219, Division for Study and Research in Education, MIT, Cambridge, MA

Carini, P. (1979). *The Art of Seeing and the Visibility of the Person*, North Dakota Study Group on Evaluation, Grand Forks, ND.

Duckworth, E. (1986). Inventing Density, North Dakota Study Group on Evaluation, Grand Forks, ND.

Evans, C., Stubbs, M., Duckworth, E., and Davis, C. (1981). *Teacher-Initiated Research: Professional Development for Teachers and a Method for Designing Research Based on Practice*, Final Report, NIE Grant # G-80-0162, Technical Education Research Centers, Cambridge, MA

Hall, G., Wallace R. C. Jr., and Dossett, W. (1973). *A Developmental Conceptualization of the Adoption Process Within Educational Institutions*, Research and Development Center for Teacher Education, University of Texas, Austin, TX.

Hawkins, J. (1985). "The Interpretation of Logo in Practice," *Technical Report No. 34*, Bank Street College of Education, New York, NY

Hull, W. (1979). *Teachers' Seminars on Children's Thinking: A Progress Report*, North Dakota Study Group on Evaluation, University of North Dakota, Grand Forks, ND

Kurland, D. M., Pea, R. D., Clement, C., Mawby, R. (1985). *A Study of the Development of Programming Ability and Thinking Skills in High School Students*, Bank Street College of Education, New York NY.

Leron, U. (1985a). "Logo Today: Vision and Reality," *The Computing Teacher*, Volume 12, Number 5.

Leron, U. (1985b) "Some Thoughts on Logo 85," *Logo '85 Theoretical Papers*, MIT, Cambridge MA.

Loucks, S. and Hall, G. (April, 1979). "Implementing Innovations in Schools: A Concerns-Based Approach." Presented at the annual meeting of the American Educational Research Association. San Francisco, CA

Martin, L. (1987). "Teachers' Adoption of Multimedia Technologies for Science and Mathematics Instruction," *Mirrors of Minds: Patterns of Experience in Educational Computing*, Pea, R. and Sheingold, K. (eds.), Ablex Publishing Corp., Norwood, NJ

Moursund, D. (1983). "Logo Frightens Me," *The Computing Teacher*, Volume 11, Number 5.

Papert, S. (1980). *Mindstorms*, Basic Books, New York NY.

Pea,R., Kurland, D. M., and Hawkins, J. (1987). "Logo and the Development of Thinking Skills," *Mirrors of Minds: Patterns of Experience in Educational Computing*, Pea, R. and Sheingold, K. (eds.) Ablex Publishing Corp. Norwood NJ.

Schoen, D. (1983). *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, Inc., New York, NY

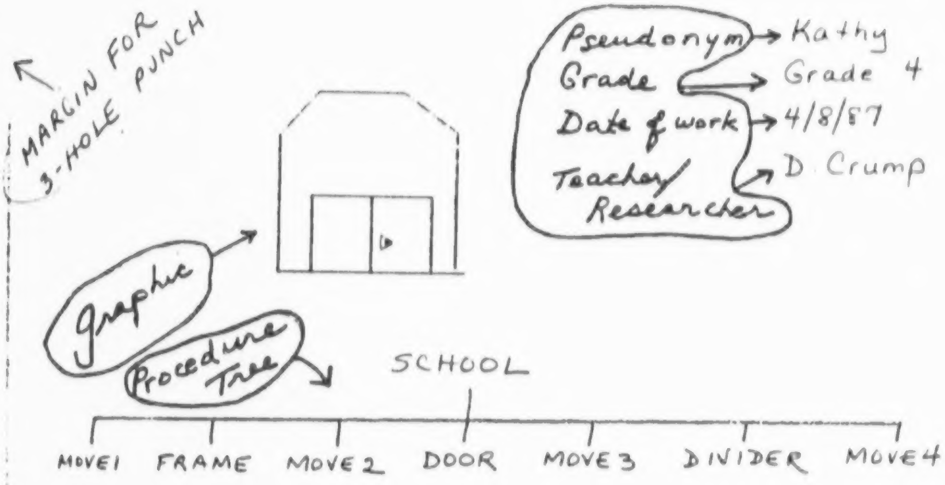
Watt, D. (1982). "Logo in the Schools," *Byte*, August 1982.

Watt, M. (1979). *Thinking About Thinking, A Description of a Professional Development Seminar*, Teacher Center, Brookline, MA

Watt, M. (1982). "What is Logo," *Creative Computing*, October 1982

Watt, M. and Watt, D. (1982). "Design Criteria for Collaborative Classroom Research, " in Amabile, T. and Stubbs, M. (eds.) *Psychological Research in the Classroom: Issues for Teachers and Researchers*, Pergamon Press, New York, NY

Watt, M. and Watt, D. (1986). *Teaching With Logo: Building Blocks for Learning*, Addison-Wesley, Menlo Park CA.



MARGIN FOR 3-HOLE PUNCH

Pseudonym → Kathy
Grade → Grade 4
Date of work → 4/8/87
Teacher/Researcher → D. Crump

Graphic

Procedure Tree

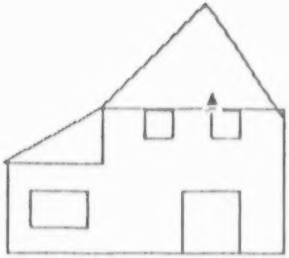
```
TO SCHOOL
 MOVE1
 FRAME
 MOVE2
 DOOP
 MOVE3
 DIVIDER
 MOVE4
END

TO MOVE1
 PU
 BK 60
 RT 90
 FD 50
 LT 90
 PD
END

TO FRAME
 FD 90
 LT 50
 FD 40
 LT 40
 FD 50
 LT 50
 FD 37
 LT 40
 FD 90
 LT 90
 FD 110
END

TO MOVE2
 PU
 LT 180
 FD 20
 RT 90
END
```

```
TO DOOR
 PD
 FD 50
 LT 90
 FD 70
 LT 90
 FD 50
END

TO MOVE3
 PU
 LT 90
 FD 35
 LT 90
 PD
END

TO DIVIDER
 FD 50
END

TO MOVE4
 PU
 BK 30
 RT 90
 FD 5
 PD
END
```

Procedures in order

FIGURE 1: Student Data Template

Kathy
Grade 4
Late March, 1987

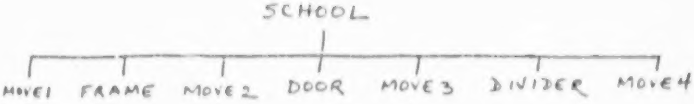


```
TO HOUSE.6
 PU
 BK 30
 BK 20
 RT 90
 FD 30
 FD 30
 LT 90
 PD
 FD 50
 FD 10
 FD 30
 LT 90
 FD 65
 FD 30
 LT 90
 FD 30
 RT 90
 FD 30
 FD 20
 LT 90
 FD 30
 FD 10
 FD 5
 FD 4
 LT 90
 FD 56
 FD 56
 FD 30
 FD 10
 FD 3
```

```
PU
LT 90
FD 56
FD 30
BK 5
BK 4
BK 3
LT 45
RT 5
RT 3
RT 4
PD
FD 20
FD 30
FD 20
FD 5
LT 103
FD 80
RT 15
FD 60
PU
RT 121
BK 15
RT 90
FD 15
PD
REPEAT 2 [FD 30 RT 90 FD 20 RT 90]
PU
FD 80
PD
REPEAT 2 [FD 20 RT 90 FD 30 RT 90]
```

```
PU
LT 90
FD 20
FD 30
FD 4
LT 90 ,
FD 5
PD
FD 15
LT 90
FD 15
LT 90
FD 15
LT 90
FD 15
RT 90
PU
FD 30
FD 15
BK 15
PD
FD 15
RT 90
FD 15
RT 90
FD 15
RT 90
FD 15
END
```

Kathy
Grade 4
Early April 1987



```
TO SCHOOL
 MOVE1
 FRAME
 MOVE2
 DOOR
 MOVE3
 DIVIDER
 MOVE4
END

TO MOVE1
 PU
 BK 60
 RT 90
 FD 50
 LT 90
 PD
END

TO FRAME
 FD 90
 LT 50
 FD 40
 LT 40
 FD 50
 LT 50
 FD 37
 LT 40
 FD 90
 LT 90
 FD 110
END

TO MOVE2
 PU
 LT 180
 FD 20
 RT 90
END
```

```
TO DOOR
 PD
 FD 50
 LT 90
 FD 70
 LT 90
 FD 50
END

TO MOVE3
 PU
 LT 90
 FD 25
 LT 90
 PD
END

TO DIVIDER
 FD 50
END

TO MOVE4
 PU
 BK 30
 RT 90
 FD 5
 PD
END
```

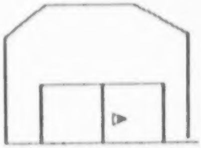


FIGURE 2: Kathy's (grade 4) work compared:

shows more sophistication in April 1987 than it did in March.

I. Procedures and Subprocedures
__________ A - Pre-procedures
__________ B - REPEAT
__________ C - Undifferentiated procedures
__________ D - Use of subprocedures
__________ E - Structured programming
__________ F - Modular programming styles
__________ G - Procedures using recursion

II - Procedures with Variables
__________ A - Pre-variables
__________ B - Procedures with simple inputs
__________ C - Procedures with proportional inputs
__________ D - Procedures with multiple inputs
__________ E - Using inputs with Subprocedures
__________ F - Procedures that involve recursion, incrementing variables and conditionals.
__________ G - Using global variables (MAKE).

III - Mathematics of turtle geometry - using 360 degrees
__________ A - 360 degrees as a complete rotation.
__________ B - Using special angles (factors of 360 and their multiples): 90, 180, 30, 45, 60

__________ C - Using angle x no. of sides = 360.
__________ D - Using the total turtle trip in constructing figures

IV - Mathematics of Turtle Geometry - other geometric ideas
__________ A - Playing turtle
__________ B - Distinguishing moves and turns
__________ C - Using inverses right/left, forward/back
__________ D - Using right/left symmetry
__________ E - Using similarity
__________ F - Using transformational geometry
__________ G - Using and understanding wrapping

V - Learning Through Exploration
__________ A - Keeping track of sequences of commands
__________ B - Taking advantage of accidents
__________ C - Looking for patterns
__________ D - Testing hunches
__________ E - Testing the limits of the system
__________ F - Making hypotheses

VI - Debugging
__________ A - Using error messages
__________ B - Recognizing familiar bugs
__________ C - Playing turtle
__________ D - Describing the bug
__________ E - Isolating the bug
__________ F - Using computer debugging tools: (PRINT, HEADING, SHOWTURTLE, TRACE

__________ G - Accepts bugs and debugging as part of the process
__________ H - Knows explicit strategies for debugging help: (simplify, look for a similar
example, ask a friend,

VII - Planning a Project
__________ A - Setting appropriate goals
__________ B - Drawing pictures as planning aids
__________ C - Dividing problem into parts
__________ D - Simplifying, setting intermediate goals, revising a plan
__________ E - Elaborating a plan
__________ F - Using a superprocedure in planning
__________ G - Using a procedure tree in planning

VIII - Managing Workspace and Files
__________ A - Saving procedures on disk in an undifferentiated workspace
__________ B - Using commands to get information: (POTS, PRINTOUT or PO, CATALOG,]
__________ C - Using record-keeping strategies to keep track of work: naming files by date or
project; recording procedure names and file names in a journal, ...
__________ D - Saving clearly differentiated files for eparate projects, special tools, ...
__________ E - Cleaning up workspace and files: erasing unneeded procedures and files

Figure 3: A draft checklist of critical aspects of Logo Learning. By making checkmarks at different places on the line, from left to right, teachers can indicate levels of student mastery, from "has barely heard of this idea," to "has mastered this well enough to teach it to others." The right hand side of the page is used for other comments.

②

```
?REPEAT 4[15 RT 90]
I DON'T KNOW WHAT TO DO WITH 15
?
?
?
?
?
?REPEAT 4[FD 50 RT 90]
?SETPC 3
?PU
?CS
```

③

```
?SETPC 0
 SETPAL 0
?SETPAL 1
?TS
?MS
?FS
?TS
?MS
?SETPC 3
?REPEAT 4[FD 50 RT 90]
?PD
?REPEAT 4[FD 50 RT 90]
?SETPAL 0
?SETPC 1
?REPEAT 4[FD 100 RT 90]
?SETPC 2
?REPEAT 4[FD 60 RT 90]
?SETPAL 1
?SETPC 1
?SETPAL 0
?REPEAT 3[FD 25 RT 120]
?FD 30
?SETPC 2
?REPEAT 3[FD 30 RT 120]
?SETPAL 1
?SETPAL 0
 FD 20
?FD 20
?REPEAT 20[FD 90 RT 33]
```

Figure 4: Sally and Pattie's dribble file (grade 5): a keystroke-by-keystroke record of their work shows that they are not using procedures at this time.

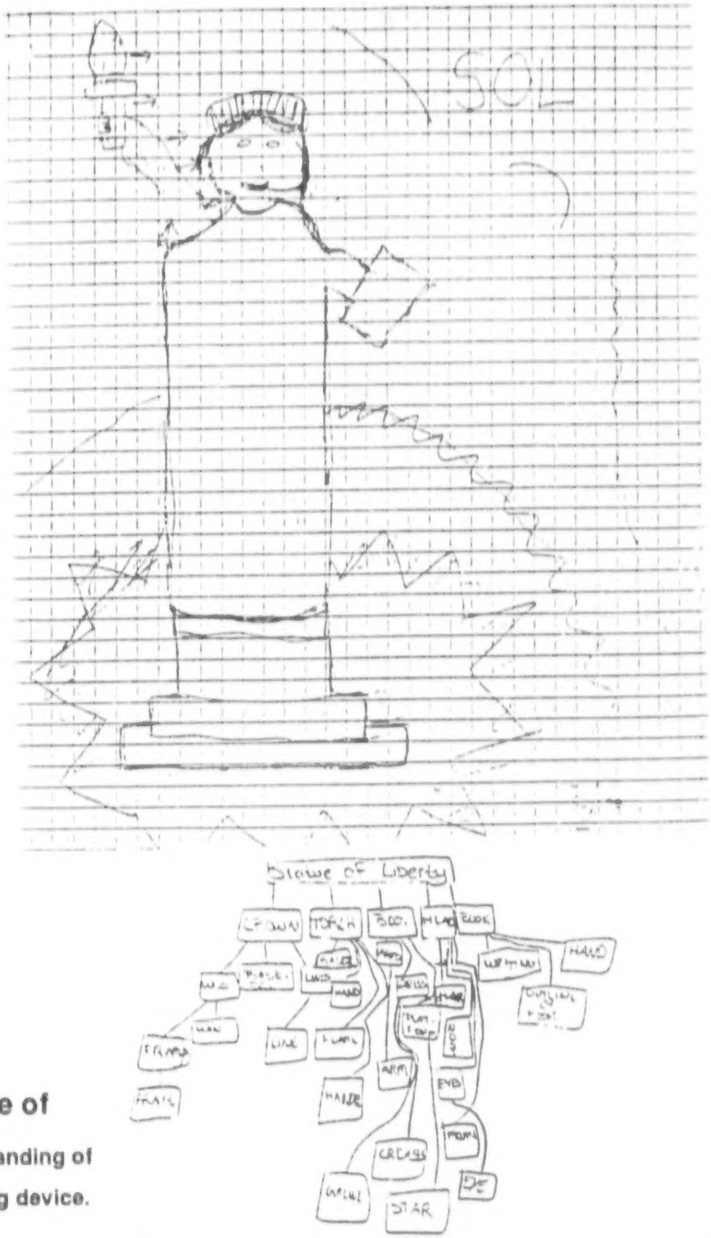

SOL
Statue of Liberty
CROWN
TORCH
BOD.
HEAD
BOOK
HAND
WRITING
OUTLINE OF BOOK
HAND
LINE
FLAME
HANDLE
ARM
CREASES
STAR
EYE
MOUTH
NOSE
FRAME

Figure 5: Heather's (grade 6) plan for Statue of Liberty project: shows that she has a good understanding of subprocedures, and uses a superprocedure as a planning device.

LOGO AND EQUITY: A NATURAL PARTNERSHIP

Sharon Burrowes Yoder
Logo Computer Systems, Inc.
330 West 58th Street, Suite 5M
New York, New York 10019

Rosemary Sutton
Department of Curriculum and Foundations
College of Education
Cleveland State University
Cleveland, Ohio 44115

Elyse S. Fleming
Associate Dean
Department of Curriculum and Foundations
College of Education
Cleveland State University
Cleveland, Ohio 44115

ABSTRACT

There are few women and minorities who hold high paying positions in today's "high tech" job market. While the number of such jobs is likely to double within the next few years, it is unlikely that the number of women and minorities in those jobs will increase because they are currently underrepresented in computer courses at the secondary and university level.

The first serious exposure to computing often occurs at the junior high level. Unfortunately, computing is usually associated with mathematics, and research clearly indicates that by the junior high age, many women and under represented minorities have acquired a high level of math anxiety. In addition, the teaching style often used in such classes usually does not involve group work, when in fact research clearly shows that group learning is preferred by girls.

In the Fall of 1987, Cleveland State University in conjunction with Cleveland area schools and Logo Computer Systems, Inc. embarked on a project designed to train teachers in group learning techniques and Logo programming. Logo lends itself well both to group work and to techniques such a graphics and sound which are also preferred by girls.

By the spring of 1988, forty teachers and five "peer trainers" will have been trained. In the summer of 1988, another group of peer trainers and teachers will be involved in similar training. At the conclusion of this project, workshop materials that include both techniques for teaching Logo and group learning will be made available.

NEEDS

Futurists predict that the number of jobs in "high tech" will double in the next ten years (Siegel, 1984). If current trends continue, this increase in jobs will not provide much hope for women and underrepresented minorities who have traditionally been trapped in low-paying jobs. In 1984, women held only 31% of the challenging and high-paying jobs as computer systems analysts but represented 91% of key punch operators and 65% of the lower-paid and dead end jobs of computer operator (U. S. Department of Labor, 1985).

Results of recent tests of high school and college students indicate that significant gaps remain in the mathematics and computer science achievement of males and females. Julian Stanley and his Johns Hopkins group report that males score significantly higher than females on the mathematics part of the SAT, the quantitative section of the Graduate Record Examination, and Calculus Advanced Placement Tests (Stanley, 1987; Brody, 1987). Test data show that males also score significantly higher than females on the computer science sections of the Advanced Placement tests and the Graduate Record Exam although males represent only 25% of those taking these exams.

The first formal study of computing typically occurs in the middle school/junior high years. It is during these years that girls' attitudes towards math deteriorate (Lockheed, Brooks-Gunn, Casserly & McAloon, 1985). Thus, these years are crucial for encouraging girls. Unfortunately most of this initial formal exposure to computers at this level focuses on the mathematical nature of the computer, further alienating girls who might otherwise choose to pursue additional courses in computing in high school. This is a particular loss if girls find their early computer experience unpleasant, discontinue study, and thus find many high paying and challenging career opportunities forever closed to them (Kersteen, Linn, Clancy & Hardyck, 1986).

The key to the problem of inappropriate presentation of computing lies in the characteristics of teachers who teach these initial courses. Most middle/junior high school teachers have little training in computer science beyond an introductory computer language course. This leads to teaching computing in a mechanistic, statement-by-statement manner. While these teachers may be aware of the greater involvement of non-minority boys with computing, they are not aware of possible strategies for dealing with this inequity. These teachers need assistance both to obtain a wider view of computer learning computer learning and computer careers and to develop new teaching strategies so that they can help non-mathematically oriented students broaden their horizons.

Seymour Papert, MIT professor and pioneer in introducing computer science concepts to children through the development of the language Logo and the book *Mindstorms* (1980), has recently concluded that teacher training is the key to preparing children for tomorrow's technological world (Papert, 1987). The Logo computer language provides a potential vehicle for increased involvement of girls and underrepresented minorities in computing. Not only does Logo make graphics, music, and text manipulation easily available, but the Logo learning environment encourages group activities. Research indicates that girls are much more engaged by the creation of graphics and music and that group learning environments are much more comfortable for female students (Schuchat, Sanders & Stone, 1986). While there is a paucity of research on the mathematics achievement and computer learning of underrepresented minorities, the available evidence suggests that the strategies that encourage girls also encourage underrepresented minorities (Sadker, Sadker, & Klein, 1986; Sutton & Fleming, 1987).

Originally, Logo was devised to provide a "math world" in which mathematical ideas could be explored in a dynamic and exploratory fashion using "Turtle Graphics" (Papert, 1986a). The "Turtle Graphics" environment that is now a part of all Logos has "helped many children bridge the arts-science gap as they found that mathematical ideas could serve their aesthetic goals on the computer screen" (Papert, 1986b). The most recent version of Logo, *LogoWriter*, provides "the material for new bridges connecting written text...into the network of all the other things children can do with their computers" (Papert, 1986b). *LogoWriter* provides a potential tool for involving girls and underrepresented minorities with the computer using techniques and learning styles they prefer while aiding them in overcoming their fears of mathematics.

More recently, a pilot project at Hennigan School in Boston suggests that girls can become involved with the computer through the use of manipulatives. In this project, *LogoWriter* was used as the vehicle to control toys made from traditional Lego building blocks. While girls' initial reactions were reserved, as the year progressed girls became more involved in the construction and control of Lego toys (Ocko, 1987).

THE PROJECT

In the fall of 1987, Cleveland State University in collaboration with the Cleveland Public Schools, Cleveland area school districts, and Logo Computer Systems, Inc. embarked on a training program designed to teach middle school teachers both *LogoWriter* and techniques for encouraging equitable use of computers in their classrooms. This project, funded by the Ohio Board of Regents through the Education for Economic Security Act of 1984, consisted of a series of Friday and Saturday workshops conducted throughout the 1987-88 school year.

The sessions were led by two instructors assisted by five peer trainers, each of whom was a practicing classroom teacher. These peer trainers had thorough knowledge of Logo programming and participated in a training session to increase their sensitivity to equity issues, prepare them to train teachers, and provide them with an overview of the project.

Each workshop focused both on equity issues and on learning Logo programming. Participants spent considerable hands-on time learning *LogoWriter*, exploring the Logo learning environment, and working on individual and group projects. Some aspect of equity issues was addressed during each workshop using techniques that research has shown to be effective in encouraging girls and underrepresented minorities to become involved with technology. At some sessions, cooperative learning environments were modeled. Role model panels and career awareness activities made up part of other sessions. One full day was devoted to working with the Logo/Lego materials.

In addition to attending all workshop sessions, participants were expected to complete a number of projects including both programming projects and lesson plans that focused on using equity techniques in day to day classroom activities. At the end of the year-long training sessions, several of the participants will be selected to become peer trainers for the sessions to be held during the summer of 1988. By the fall of 1988, this multiplier model will result in the training of approximately 80 teachers and in leadership training for another ten.

The success of this project is being evaluated using both formative and summative techniques. Evaluations are given at the end of each workshop to assess the success of individual activities. Debriefing sessions with peer trainers and instructors further examine each activity, with a particular focus on anecdotal information. At the conclusion of the project, a larger scale analysis of the effect of the project on the participants and their students will be completed.

During the development of this project, a series of workshop notes have been developed. These notes, including information on how to present Logo workshops and techniques for enhancing the involvement of under represented minorities and girls in the use of computers, serve as a record of the workshops and provide guidelines to others wishing to offer a similar series of workshops in the future. These materials will be made available for dissemination at the termination of the grant in the fall of 1988.

APPLICABILITY TO THE CLASSROOM

The techniques used during this teacher training project to teach teachers about the Logo learning environment and to encourage equitable use of the computer have been used successfully in a variety of environments and are equally useful in a classroom of middle-school students. Collaborative learning, a technique all too often overlooked by the traditional classroom teacher, encourages cooperation and learning by all.

The Logo learning environment encourages creativity, discovery, and appropriation of material by students. Girls respond positively to graphics and sound generation. Both of these are an integral part of the *LogoWriter* environment. Using Logo in a manner that is supportive of different learning styles is both natural and powerful and can result is much more success and positive feelings on the part of girls and under represented minorities.

FUTURE

This project can serve as a model for others wanting to train teachers in equitable use of the computer in the classroom. Teachers learning both Logo and equity techniques return to their classrooms with ideas that will encourage all students to make the computer an integral part of their day to day lives.

Clearly, computers are going to be a part of everyone's future and it is therefore essential that we take seriously the training of each young person in our schools. Students who are comfortable with computers and computing are less likely to reject job opportunities that involve technology. Only if they feel comfortable with this emerging technology will every person, whatever gender or race, be able to take full advantage of the society into which they will graduate.

REFERENCES

Brody, L. E. (1987, April) "Sex differences on the GRE, MCAT, LSAT, and GMAT." Paper presented at the annual meeting of the American Educational Research Association, Washington, DC.

Kersteen, Z. A., Linn, M. C., Clancy, M., & Hardyck, C. (1986, April). "Previous experience and the learning of computer learning: The computer helps those who help themselves." Paper presented at the annual meeting of the American Educational Research Association, San Francisco, CA.

Lockheed, M., Brooks-Gunn, J., Casserly, P., & McAloon, A. (1985, May). "Sex and ethnic differences in middle school mathematics, science, and computer science: What do we know?" Princeton, NJ: Educational Testing service, A Report submitted to the Ford Foundation.

Ocko, S. (1987, May). "Learning as a reconstruction of knowledge: The Lego/Logo concept." Keynote address of the Great Lakes Logo Conference, Cleveland, OH.

Papert, S. (1986a, April) "The next step LogoWriter." *Classroom Computer Learning*, 38-40.

Papert, S. (1986b). *From the desk of Seymour Papert.* A letter to LogoWriter licensees.

Sadker, M., Sadker, D., & Klein, S. (1986). "Abolishing misperceptions about sex equity in education.." *Theory into practice*, *25*, 219-226.

Schuchat Sanders, J., & Stone, A. (1986). *The neuter computer*. NY: Neal Schuman.

Siegel, L. (1984). "Researchers challenge: Portrait of a high-tech future." *Infoworld*, *5*, (20), 31.

Stanley, J. C. (1987, April). "Sex differences on the college board achievement tests and advanced placement examinations." Paper presented at the annual meeting of the American Educational Research Association, Washington, DC.

Sutton, R. & Fleming, E. S. (1987, February). *EQUALS at Cleveland State University: 1985-6 evaluation report.* Cleveland, OH: Cleveland State University.

U. S. Department of Labor. (1985). *Employment and unemployment during 1984: An analysis.* U. S. Department of Labor, Bureau of Labor Statistics.

SECONDARY CONTENT AREAS

Jon Young, Cliff Hardy
Kathleen Maury
Susan Merritt

ABSTRACT: The Texas Learning Technology Group--A Physical Science Interactive Videodisc Curriculum Project

Dr. Jon Young, Department of Computer Education and Cognitive Systems, North Texas State University; and Dr. Cliff Hardy, Department of Secondary Education, College of Education, North Texas State University, Denton, TX 76203

Formed in 1985, the Texas Learning Technology Group is coordinated by the Texas Association of School Boards, initially funded with a $500,000 grant from the National Science Center Foundation and an additional $1.5 million contributed by 12 Texas school districts.

The efforts of this group have resulted in the development of a complete technology-based delivery system integrating computer, video, and audio technology programs. Important aspects of the program include teacher training and involvement, the involvement of higher education staff in curriculum review and teacher training, and thorough coursework evaluation plan.

The electronic-based curriculum will span 160 hours, or two semesters, of physical instruction, with the teacher leading the students through small group and laboratory activities. The role of the teacher is shifted in the process to that of a classroom manager, as opposed to the traditional performer. The teachers are being trained on an ongoing basis in a process that will ultimately be assumed by selected Texas universities that will offer courses in videodisc application.

The laserdisc computer program utilizes the IBM Info-Window touch screen application and combines the qualities of a television program with the capabilities of a computer. The curriculum is packed with unusual scenarios designed for student motivation but yet presents a firm foundation in the physical sciences. For example, when students want more in-depth information about a scientific concept, or fail to grasp a concept, the computer program allows them to see the scientific principal approached from a different angle by a different narrator and different illustrations. In addition, the program allows students to work at their own pace. Slow students each advance through the basic work and if they still do not understand, they can ask the computer program for another run-through. The touch-information windows will also allow handicapped students to activate the program with a pencil or pointer. An infrared grid overlay, rather than heat sensors, makes it possible for handicapped youngsters to control their learning and even conduct scientific experiments.

The program is now in its first year of field testing in selected Texas schools and the project is undergoing a thorough and realistic evaluation. The physical science courseware and other courseware developed by the group will be evaluated for effectiveness in two ways. First, the Texas Education Agency will conduct independent evaluation of the effectiveness of the courseware. Second, the Texas Learning Technology Group has established an evaluation methodology based on the input-output model developed by the University of Texas Educational Productivity Council.

ABSTRACT: The Development of an Instrument to Measure the Impact of Calculators and Computers on the Secondary School Mathematics Curriculum.

Kathleen Maury, Assistant Professor of Education, Division of Education, University of Wisconsin - Superior, Superior, WI 54880

An instrument to measure the impact of the calculator and the computer on the secondary school mathematics curriculum was developed and tested. In the process, information about the variables which explain the impact of calculators and computers on the secondary school mathematics curriculum was gathered, and information about the specific impact that calculators and computers have had on the mathematics curriculum was collected.

In order to complete the analysis, the respondents were catagorized into two groups: the high impact technology users and the low impact technology users. The analysis of these impact groups indicated that hardware and software accessibility, the teacher, and the curriculum requirements significantly differentiated between the high impact technology using group and the low impact technology using group.

The high impact technology using teachers reported adding mathematics concepts to their curriculum while they reported dropping the study of arithmetic algorithms from their curriculum. The data suggested that these teachers increased the importance of mathematics concepts in their curriculum while they devalued the importance of the arithmetic algorithms. It could be said that they devalued the importance of those algorithms that their machines could easily execute.

In addition, technology using mathematics teachers reported teaching different skills. These data suggested that these skills were developed simultaneously with the concept rather than before or after the concept. Further, these data suggested that topics typically included within other disciplines become integrated into the mathematics curriculum when mathematics teachers consistently use technology. Finally, technology using mathematics teachers reported that 48% of their computer lessons were individualized as opposed to large group or small group lessons.

ABSTRACT: Composition Approach to Word Processing

Susan Merritt, Business Education Teacher, Dimmitt High School, Dimmitt, TX

In order to more adequately prepare our students for the business world into which they will move, I suggest that word processing be taught from the composition approach. For years we taught the students how to take shorthand dictation, but never taught the management level how to give dictation. In my own personal business experience there was only one man out of ten who could give dictation. More often than not instead of a secretary, management now finds a microcomputer at his desk. They need to learn the skills necessary to take advantage of the microcomputer. They need the skill to compose their letters, reports and documents directly at the computer. The computer is here to stay in the business world and we need to prepare the students to make the most efficient use of it possible. Time is money and keying in a document directly is much more efficient than writing a pen and paper rough draft and revising and then passing it to a clerk to be typed. The computer and word processor can make quick work of any document - as long as the brain controlling it is trained in the most efficient use.

Composing at the keyboard is not a natural skill. It is one that must be acquired. I have developed a word processing teaching unit that incorporates the use of composition at the keybard with the introduction of the word processor. It in no way adds any burden to the student in the learning of the particular word processor. The students aren't even aware that they are learning to compose at the keyboard.

SPECIAL NEEDS AND EQUITY: MIDDLE-HIGH SCHOOL

Darlene Grantham
Richard Smith, Ellen Murphy
Linda Ware
Ricki Price

ABSTRACT: Sex Equity Initiatives in Montgomery County Maryland

Darlene Grantham, Teacher Specialist, Computer-Related Instruction, Montgomery County Public Schools, Rockville, MD 20850

The Montgomery County Maryland Public School System has adopted a sex equity initiative goal for grades 7 - 12. Staff in Academic Skills, Computer-Related Instruction, and the Guidance office at the county level have been addressing the goal:

> *To encourage greater female enrollment in computer science and advanced mathematics and science courses, and improve performance by females on the mathematics section of the S.A.T.*

School system plans and progress in meeting the goal are summarized below.

S.A.T. materials were developed and placed in all schools for use in Geometry and above mathematics classes. This added emphasis on the S.A.T. may have contributed to the improvement in the S.A.T. scores this year. Scores for females on the mathematics section increased an average of five points, from 495 to 500.

In May, 1987, a conference was held to provide awareness of the issues related to sex equity. Each county secondary school sent a team of a mathematics, science, and computer science teacher; a guidance counselor, media specialist, and career technician; and an administrator, two parents, and two students. The conference agenda included: a panel of speakers representing industry, government, the armed services, and education; a preview of selected videotapes and films on sex equity; a biographical sketch of a famous female mathematician; and startling statements to increase awareness of statistics involving sex equity. School teams discussed implications of materials presented and developed a list of desired resources.

As a follow-up to the spring conference, a summer workshop was held to begin the development of activities suggested by schools and the identification of resources to be purchased for sex equity. These activities and resources comprise the Montgomery County Sex Equity Resources Book.

Training on the Resources Book began in October, 1987, at a workshop attended by over 300 school system staff. These staff members are involved in the pilot of the use of the materials. They are training additional staff in their respective discipline and providing feedback on their progress, both successes and failures.

In May, 1988, a follow-up work session was held to make recommendations for modifications in the Resources Book, to report on progress made during the 1987-88 school year, and to plan the next steps including future strategies involving county elementary schools.

Contact People: L. Joy Odom, Secondary Mathematics Coordinator; Grace Tigani, Secondary Science Teacher Specialist; Bernice Ryder, Secondary Guidance Teacher Specialist

ABSTRACT: Two Projects Using Technology-Assisted Instruction To Teach Academically At-Risk Students

Richard Alan Smith, PhD; Manager, Instructional Development Group; and Ellen R. Murphy, Technology Trainer, Instructional Development Group, Bureau of Technology, Houston Independent School District, Houston, TX 77056

The Houston Independent School District (HISD) 1986 Summer TEAMS Academics project was a special project designed to improve the performance of academically at-risk eighth grade students on the Texas Educational Assessment of Minimum Skills (TEAMS) test. Funding from a Chapter 2, Desegregation Assistance Grant, that had the goal of improving the performance of academically low-achieving minority students, provided the unique opportunity to capitalize on the talents of enthusiastic teachers, the capabilities of educational technology and a wealth of commercially available curriculum materials in order to improve student test scores.

The project plan was based upon the following:

1. The teachers and program administrator would be given as much professional freedom as

possible, within the boundaries of the project goals, to design the daily lessons and activities for the students.

2. In addition to learning subject matter the students would be taught test taking skills.

3. The students would also be taught study skills and be motivated to continue good study habits throughout the new school year.

4. Enough time would be alloted for the classes to insure that each student received the maximum impact possible from the project.

5. The classes would take place in computer classrooms and the teachers would be supplied with all necessary materials and appropriate educational technology.

The final results of the eight-week Academics surprised even the project designers. Those students who completed the program passed the TEAMS test at a rate higher than the general HISD population and a rate far above what teacher experts would have predicted for their fellow at-risk students.

Of a total of 14 at-risk students who completed (attended sessions on a regular basis) the 1986 TEAMS Academics, 78.6% passed the math section, 85.7% passed the reading section and 71.4% passed the writing section. All ninth grade HISD students had a pass rate of 77% for the math section, 73% for the reading section and only 57% for the writing section. Even the at-risk students who participated in the Academics but did not attend sessions on a regular basis (Academy noncomplete) had a pass rate almost equal to all HISD students. Their pass rates were, math 64.3%, reading 71.4% and writing 64.3%. In addition, the TEAMS Academics students had a higher pass rate when compared to a matched group of students who did not participate in the Academics. The matched group students had TEAMS passing rates of 68.2% mathematics, 72.7% reading, and 40.9% writing.

An outgrowth of the 1986 Summer TEAMS Academics project has been the development of a curriculum to be used in tutorials for students who have already failed, or who are deemed in danger of failing, the exit level TEAMS test. This curriculum involves the use of software, print materials, and video to teach and reinforce the eighteen objectives on each of the exit level tests (mathematics and language arts). Each unit includes:

1. Correlations to district objectives;
2. An instructional focus detailing prerequisite skills;
3. Computer, print, and video activities;
4. Instructional strategies; and
5. Progress checkpoint activities.

These curriculum materials can be used in formal or informal tutorial sessions, as well as a supplement to regular or remedial mathematics or language arts classes. Development plans include formulation of additional curricula to address the TEAMS objectives at the lower grade levels.

ABSTRACT: Project TEMPT - Teaching Explorations with Mind-Powered Technology

Linda Ware, MEd, Project Director/Resource Teacher, Ysleta Independent School District, El Paso, TX

The Ysleta Independent School District (60,000 students), has committed its middle school resource teachers to Project TEMPT, a three-year training project, using computers with an established focus on problem-solving software and tool applications. The training of the 29 teachers will include the necessary curriculum revision technology promotes, teaming of regular and special education teachers, and the generation of off-computer activities which incorporate across-the-curriculum learning objectives.

The project designed by this presentor is based on six years of classroom instruction utilizing computers with LD and ED students to facilitate improved concept development in the area of mathematics and language arts. Of critical importance to the curriculum is the opportunity for improved communication among students and a teamwork approach to learning. As a result, students' self-concept is strengthened while the traditional stigma of resource placement is minimized by the unique opportunity this project provides to integrate computers throughout their instruction.

In our resource rooms, students use the computer for a variety of activities ranging from the classification of shapes in a Venn diagram using *Gertrude's Puzzles*, to the creation of a spreadsheet using *AppleWorks* to keep a file consisting of grades from their own classes. Off-computer activities include the generation of a Venn diagram to assign vocabulary words from their regular education classes which reinforce the classification learning strategy beyond the characters and shapes presented by the software. In another off-computer activity, students use the spreadsheet printout of their own grades to make predictions about the final grades they will earn. Frequently, students realize they are in danger of failing as they see their grade point generated on the spreadsheet, and take last minute action to avoid a failing grade.

Teachers follow a purposely flexible curriculum which encourages each teacher to select which software will be utilized and order the activities according to the priorities established at each of the different campuses involved. In the same way that students gain responsibility for their own learning by making decisions about their work, so too, do the teachers involved in this project as they have the opportunity to decide the focus of the computer's use in their classroom. Bimonthly meetings with the project teachers keep the participants updated on student projects, teaching strategies, and the application of technology their colleagues have attempted.

In addition, the meetings serve to reassure project teachers that integration is a slow-process which must be conceptualized across time as well as across subject matter. Consistent with one of the characteristics of problem-solving software: "Errors are necessary and unavoidable"-- so too are our initial steps in project TEMPT as we problem solve our plan for a meaningful computer curriculum for middle school resource students.

Project TEMPT Recommended Software:

- *Moptown Parade/Hotel* (The Learning Company)
- *Gertrude's Secrets/Puzzles* (The Learning Company)
- *The Pond* (Sunburst)
- *Challenge Math* (Sunburst)
- *Survival Math* (Sunburst)
- *Building Perspectives* (Sunburst)
- *The Factory/Super Factory* (Sunburst)
- *Memory Castle* (Sunburst)
- *Incredible Lab* (Sunburst)
- *Discover* (Sunburst)
- *Puzzle Tanks* (Sunburst)
- *The King's Rules* (Sunburst)
- *Suspect Sentences* (Ginn)
- *Mystery Sentences* (Scholastic)
- *AI* (Scholastic)
- *LogoWriter* (LCSI)
- *AppleWorks*

ABSTRACT: ESL for Preliterate Middle-School Students

Ricki S. Price, Project Management Coordinator,
Houston Independent School District, Houston, TX

In 1986-87 the Houston Independent School District's Multilingual Services Department began a project to use computers as an integral part of the English as a Second Language (ESL) curriculum for preliterate students in 10 middle schools. The program was expanded to five additional campuses the following year.

The challenge was to select software to enhance teaching the most basic levels of English reading and writing without being too childish for the middle-school student. Project planners found 21 commercially available packages that allowed creativity and provided the opportunity to incorporate listening, speaking, reading, and writing activities.

One printer and four computers equipped with speech boards were placed in a single classroom at each of the 10 sites. The district's Bureau of Technology trained each of the project's ESL teachers. The 26 hours of training included how to infuse the technology into the ESL curriculum. Additionally, consultants were assigned to visit each school regularly to provide on-site support.

Teachers reported that student behavior, attitude towards school, attendance, and achievement improved. Using computers required teachers to change their teaching methods and use more grouping strategies, but most agreed that the results were more than worth the effort. Many of the ESL teachers who had been hesitant about using technology began to lobby for additional computers.

The success of any project of this kind rests with the teachers. The training and on-going support that teachers receive is critical.

INTEGRATING TECHNOLOGY INTO THE TEACHING OF STATISTICS AND DATA ANALYSIS (GRADES K-12)

Bertram Bruce
Reasoning Under Uncertainty Project
BBN Laboratories, Inc.
Cambridge, MA 02138

Susan Friel
Used Numbers Project
Lesley College
Cambridge, MA 02238

ABSTRACT

The newspapers are full of articles that use statistics, and everyday activities such as buying a stereo or choosing a healthful diet call for sound probabilistic and statistical thinking. Moreover, much of scientific work is crucially dependent upon statistics. Yet statistical reasoning is seldom taught to pre-college students, and when it is, the emphasis is often on formulaic manipulation, rather than on the concepts that are the foundation of reasoning about statistical matters.

Two projects, both funded by the National Science Foundation, address these concerns. This presentation will provide an overview of each of these two projects and demonstrate some of the applications of technology that have been integrated into the curriculum.

Used Numbers: Collecting and Analyzing Real Data, funded by a grant to TERC, is developing curriculum modules for grades K-6 which focus on data analysis-collecting, organizing, graphing, and interpreting quantitative data. The modules emphasize central concepts in statistics, estimation, and graphing, the application of computation in meaningful contexts, and the appropriate use of calculators, graphing software, and spreadsheets.

TERC, Lesley College, and the Consortium for Mathematics and Its Applications (COMAP) are collaborating under this three-year grant. The project will develop a curriculum module for each grade level, pilot and field test these modules, produce video and print teacher training materials, devise measures of student learning of critical ideas in data analysis, and make recommendations about developmentally appropriate understanding of these ideas in the elementary school.

Reasoning Under Uncertainty, funded by a grant to BBN Laboratories, Inc., has developed a computer-enhanced curriculum in statistical reasoning for that incorporates a software system called ELASTIC(TM). The course is designed to help high school students develop statistical reasoning abilities by using real world activities with which they have practical experience.

The ELASTIC(TM) software, implemented on a Macintosh computer, is a tool for recording, representing, and manipulating statistical information. It has a spreadsheet-like interface to a database that can represent either categorical or numerical variables. Students can easily display summary statistics and create graphs, such as histograms, box plots, scatterplots, and bar plots. ELASTIC(TM) also contins three interactive programs: *Stretchy Histograms*, *Sampler*, and *Shifty Lines*.

The curriculum and software were field-tested in two high schools in the spring of 1987. The field test provided support for the basic curriculum and software approaches, suggestions for extensions to both, and new insights about the processes of learning and teaching statistics.

PUTTING LONGEVITY INTO THE TEACHING OF APPLICATION SOFTWARE

Tony Jongejan
School of Education
Western Washington University
Bellingham, WA

ABSTRACT

In the 1970's the cry went up for the Teaching of Computer Literacy. With the development of the microcomputer and its subsequent inclusion into K-12 Education, this cry only became louder. Students obviously needed to be prepared for an increasingly computerized world. School, districts and states went as far as to require Computer Literacy courses. Ongoing questions such as "What does computer literacy mean?" and "What should be included in a Computer Literacy experience?" were largely ignored or, at least, minimized. Many educators felt that computer literacy should at the very least include an extended experience with computer programming.

In the early 80's, this outlook quickly changed. Schools began replacing the programming experience with time spent learning to use application software. The cry changed from "To be Computer Literate, you need to know how to program." to "In the real world, people don't program; they use previously prepared or 'canned' software. " Beyond the very important question of whether this replacement of programming was appropriate, two other points were largely overlooked.

First of all, most 'application' classes dealt with only three applications: word processing, data bases and spreadsheets. Those were the three areas where business was using computers and they were seen as the areas that would have the most immediate practical applicability for the student. Software was even developed that integrated these three applications into one package. But, Computer experiences such as this should prepare students for the future. The applications of tomorrow may not look at all like those of today. Already new applications such as desk top publishing, telecommunications and computer assisted design are becoming increasingly important.

Secondly, the teaching of applications largely became the teaching of how to get a specific piece of software to perform a specific task on a specific computer; it became the teaching of key strokes rather than the teaching of the general concepts involved. For example, students learned the key stokes to do such things as double spacing text using FrEdWriter on an Apple Computer rather than the concept that the line spacing of text is an area that word processors usually handle in one of three ways. (86)

Thus, a change of emphasis needs to occur. The focus of these 'application' experiences needs to be on the concepts involved, not on the keystrokes to be used. First, general concepts that cross over all examples of applications need to be covered. Ideas such as the computer being a 'tool' to accomplish needed tasks and that application software converts a generic computer into this tool need to be emphasized. (65)

Next, each application has a number of key concepts that need to be understood. If the student understands them, then they can transfer their knowledge to other pieces of application software and/or machines and, eventually, to other types of applications. For example, examination of the most popular spreadsheet software shows that almost all have three lines at either the top or the bottom the screen that serve very similar functions: (1) The entry line which shows the contents of the current cell, (2) The edit line where cell contents are entered and edited and (3) The prompt line which gives the available options at any stage in the users work.

Finally, students need an extended experience with one software package on one computer so they become comfortable applying the concepts being taught. At the same time, students need a variety of experiences on a variety of application software packages and a number of different microcomputers so they can see the similarity in the way these concepts are handled.

This presentation will discuss these ideas further focusing on (a) the concepts that cross over all applications and (b) the concepts that need to be emphasized when teaching students to use word processing, data base and spreadsheet software. Examples from specific software packages and computers will be shared.

RESEARCH AND EVALUATION TRENDS IN THE USES OF COMPUTER-BASED TOOLS FOR LEARNING AND TEACHING

Beverly Hunter
Targeted Learning Corporation
Route 1, Box 190
Amissville, VA 22002
(703) 937-5500

ABSTRACT

Use of computer-based tools to support the learning process is increasing rapidly in elementary and secondary school classrooms. The reasons for their use range from a straightforward desire to expose children to the tools of the information age, to more ambitious goals of reforming the basic curriculum and pedagogy to make it more process-oriented and reflective of real-world complexity. This paper reports on our changing knowledge regarding the effects and conditions of tool uses of computers in learning and teaching as reflected in current research and evaluation studies. We identify strategic areas for research, development and evaluation.

INTRODUCTION

Tool applications are the fastest-growing genre of computer application to instruction in elementary and secondary schools. Yet we are just beginning to understanding whether or how which tools may help which students under what conditions.

The purposes of this paper are to: 1) define and summarize the current state of the art of tool applications of computers for learning and teaching in elementary and secondary education; and 2) suggest strategic areas for research, development and evaluation of such uses.

CHARACTERISTICS OF TOOL APPLICATIONS TO LEARNING AND TEACHING

A computer-based tool is a program a person uses to perform or facilitate an intellectual task, analogous to the way one uses a screwdriver or hammer to facilitate a physical or mechanical task. Key characteristics of a *tool* are:

- it is under the control of the person performing the task;
- it can be used to accomplish a variety of goals and tasks for a variety of purposes;
- the user determines the goals and purposes of the task at hand;
- the skills and intentions of the user determine the quality and nature of the process and product.

These characteristics distinguish uses of computer-based tools for learning from use of the computer as a teaching machine or object of study.

Why Tools Are Used: Pedagogical Philosophy

Tools are usually used to support process-oriented learning including inquiry, writing, problem-solving, decision making, collaboration, and other applications of higher-order thinking skills. They are also used to support concept learning and to provide access to information and algorithms.

Although educators have many reasons for providing tools as part of learning environments, there are certain themes of pedagogical philosophy that underlie much of the work with tools (e.g. see Nickerson [103]; Burton [22]; Papert [105]; Tucker [129]; Patterson & Smith [108]). These themes include:

- Constructivism. The learner is viewed as an active constructor of knowledge. The learner is not an empty vessel into which knowledge is poured; rather, the learner must be active and must be relating new information to existing knowledge (e.g. Hunter, [65], [67]).
- Conceptual understanding. It is believed inappropriate to have students learn procedures by rote. The concepts underlying procedures must be taught or discovered if the student is to be able to apply procedures to situations beyond those taught. (Brown *et al.* [16]; Brown *et al.* [18]).
- Self monitoring and self management of learning. Learners are viewed as taking responsibility for their own thinking and learning activities. This implies the need for metacognitive skills such as planning, directing attention, assessing their own comprehension and controlling anxiety.
- Lifelong learning. Education must be an ongoing process throughout life, not something acquired in school and later applied.

Use of Tools is Growing Rapidly

As more teachers become computer literate and more appropriate software tools become available, the tool use of computers in learning and teaching is growing rapidly. Many articles and books on the subject of using tools have been published recently (e.g. Canning *et al.* [24]; Hodges [58]; Hunter [62], [63]; Lockheed & Mandinach [77]; Sopp [120]; MicroSIFT [5], [132]). The State of North Carolina developed data bases on counties in their state to support the junior high course on North Carolina history, and distributed them to all 8th grade teachers in the state [104]. Teachers in school districts throughout the country developed data files and learning activities to support social studies curricula (e.g. [95]; [35]; [136]; [53]; [23]; [122]). During the past three years, approximately thirty data base packages to support the elementary and secondary school social studies curriculum were published by educational publishers. The Johns Hopkins

University 1985 national survey of instructional uses of computers asked the primary computer-using teachers which one of three major uses of computers seemed best at the time their school got its first computer and which one seemed best in 1985. The computer as a tool of the learner started as a small percentage of the priority of computer using teachers, but it is growing rapidly in importance relative to the two other uses shown [7].

In a survey, Talmis [123] found the largest percentage of their school respondents (27%) expected the use of the computer as a tool for students to use in writing and analyzing data to increase more rapidly than other uses. Schools with five years or more of computer-using experience are more likely to select tool use than are schools new to computing. Although the most frequent type of software *owned* by the schools was CAI drills and tutorials, the most frequently *used* programs were tools.

Research on tool uses has been supported for several years by Apple Computer [12], the National Science Foundation and other funding agencies.

NATURE OF CURRENT STUDIES OF TOOL USES

Current research and evaluation in this field is diverse, reflecting the wide variety of tools and purposes for their use. However, certain general characteristics distinguish this genre of research from earlier work with computer-assisted instruction.

The Research Focuses on Many Dimensions of Learning, Learners, and Learning Environments

Current research and evaluation in tool use seeks deeper understanding of the complex interaction of children, tools, and classroom learning environments. Earlier research in educational computing typically asked questions like "Does the computer make a difference in learning x?" Current tool researchers seek to be more sensitive to the multidimensional nature of learning processes, learning outcomes and learning environments. It is becoming more commonly understood that *technology is not an independent variable* -- that the ways in which a given tool are used depend on the cognitive, affective and social skills of learners as well as the social environment of the classroom, the school, and educational institutions (e.g. Michaels [88]).

Researchers Are Becoming More Interested in Individual Learners' Needs

Rather than asking group outcome questions, researchers are becoming more interested in the usefulness of particular tools or particular features of those tools, for particular learners. For example, Rosegrant [118] is interested in which features of the Talking TextWriter are used by children at various ages and developmental levels in their acquisition of literacy.

Tool Researchers are Interested in Processes as Well as Outcomes

Since tools are most often used to help in learning processes, the focus of the research and evaluation is correspondingly more process-oriented (e.g. Fisher [49]). For example, many studies of the role of word processors in writing are focused on the interaction of the tool with the nature of the writing process (e.g. Bruce, Michaels & Watson-Gregeo [19]; Daiute [33];). Similarly, the impact of the tools on science processes is of interest (e.g. Berger [8], [9], [10]). Many of the teachers and researchers working on tool uses are more interested in cognitive processes than in behavioral outcomes (e.g. Anderson *et al.* [2]; Barclay [4]; Collins & Gentner [30]; Dervin [34]; Fisher [49]; Hunter [65], [67], Malone [80]; Perkins [110], [111]; tum Suden & Rowe [131]).

Some Research Focuses on the Effects of Knowledge Representation on Learning and Understanding

Tool researchers are interested in interactions between learner and ways in which knowledge is represented by the tool. Thus the question is not simply "Is this tool effective in learning x?" but more subtly, "How does this particular characteristic of the knowledge representation help or hinder the learning process ?" (e.g. [47], [22], [2], [71]). For example, many studies are now being made of children's use and understanding of graphs in microcomputer-based laboratories (Brasell [14], [15];Linn *et al.* [8]; Mokros & Tinker [94]; McKenzie & Padilla [83]).

Some Research Focuses on Teachers' Roles

Teachers are inventing and creating new roles for themselves as they introduce new tools and processes into their classrooms (for example, more coaching and less lecturing). The roles of teachers are becoming an increasing focus of attention in our attempts to understand the uses of the tools in learning (e.g. Collins & Stevens [31]; Cohen et al [26]; Copeland [32]; Hawkins & Sheingold [57]).

Collaborative Learning is an Increasingly Important Focus

Relationships among learners using a tool are of increasing interest to researchers as well as teachers, partly because collaborative learning is often a major goal of the tool-using instructional innovation. Collaboration among students in small groups has been demonstrated to be a more effective mode of learning that either competitive or individualistic modes (e.g., Johnson, Johnson & Stanne [70]).

Evaluation Methods are a Major Concern

Often, tools are introduced to enable students to address more complex problems or concepts than they could without computer-based tools, or to focus on learning processes in new ways. A major concern then becomes how to assess these new learnings when existing tests and other assessment instruments were not designed for the educational purpose at hand. White [133], [134] and Parker [107], for example, spent a major part of their research efforts on the development of instruments to measure information processing skills. Layman & Kirkpatrick [72] devised a laboratory practicum to test students' microcomputer-based laboratory skills.

The Catch-22 of Tool Uses in the Existing Curriculum

Advocates of uses of computer-based tools in elementary and secondary education face a seemingly unbreakable cycle of dilemmas. Use of the tool must support the existing curriculum and textbook if teachers are to adopt the innovation. But the existing curriculum and textbook were designed to meet the goals and needs of a paper-and-pencil print-oriented world. Hawkins & Sheingold [57] report on teachers' experiences in attempting to use general-purpose data management programs in science and social studies.

> Curricular issues were prominent in these two programs of research because we chose to study uses of the technology that had the promise of supporting skills of problem solving and critical inquiry. As such, the software did not have an obvious fit with the traditional curriculum. The curriculum had to be stretched or modified in some way to accommodate the technological functions...We were interested in how such stretching of the standard curricular vision, in the direction of higher-order problem solving skills that will likely be prominent in an information age, would be accommodated by today's classrooms [57].

EXAMPLES OF THE TOOLS AND STUDIES OF THEIR USES IN LEARNING AND TEACHING

The following tool applications and research on their uses are illustrative of current work in this field. Dozens of equally important efforts are not mentioned due to space limitations.

Tools Are Used Most Often to Support the Writing Process

The most commonly used computer-based tools in elementary and secondary education today (excluding programming languages) are used to support the writing process [28], [52], [138]. Word processors and related software are used in a wide variety of ways to support development of skills in the writing process -- from prewriting, to writing, to editing and publishing one's work. In 1985, among high school English teachers using computers, fully 69% indicated that students had improved their writing, editing or proofreading skill by using computers -- a greater consensus than for any other "effect" of using computers for any subject at any grade level. In addition, more business education teachers indicated improvements in student writing than indicated any other learning effect from computer use (21%). Overall, 19% of all high school computer-using teachers suggested that improved writing was an important outcome of their use of computers, a higher percentage than those naming any other outcome [6], [7].

Studies on using computers to help improve writing skills involve a wide variety of environments and evaluation criteria. After several years of experience, it is becoming more commonly understood that simply using the technology of word processors does not in and of itself improve writing skills of students or the teacher's ability to teach writing skills [81]. Daiute's book, *Writing and Computers* [33] examines the computer in writing instruction. She proposes that the introduction of the computer into the writing process changes the nature of the activity because it provides the writer with more power. In one study, Daiute concluded that the positive effects of computers on writing were attributable to recent changes in writing philosophy and approaches to using computers in education as much as they are related to the actual use of the computer itself. In a series of studies at the Interactive Technology Laboratory at UC San Diego, the computer was found to provide students with greater freedom to express their ideas by facilitating the invention, revision and publication of their compositions. Researchers have examined effects on student writing and editing resulting from functional learning environments (e.g. developing a newswire service or communicating via computer networks). These studies indicate that the development of functional writing environments, which contextualize students' learning activities in the classroom, improve the quality of students' writing. (Cohen & Riel [27]; Riel [113], [114]).

Other studies have focused on differences among students in their uses of the computer in writing. One study [137] concluded that poor writers made editing changes in their writing only at the word level, whereas better writers considered the overall effect of their document, inserting and deleting large segments of text before searching individual sentences for word-level errors. Morocco [96] observed that in remedial settings teachers overemphasized the editing capabilities of word processing software, interrupting the writing process too soon for editing.

Specially designed tools for children support the full range of the writing process. The QUILL project, for example, addresses six main goals [119]:

- Help children develop skills in planning and critical thinking.
- Integrate reading and writing.
- Make writing public.
- Support communication with real audiences.
- Encourage writing with and for peers.
- Facilitate revision.

Each goal is supported by a set of tools such as note takers and an electronic messaging system. In their research with children using these tools in classrooms, investigators found that the tools alone do not account for the most important changes which take place. Rather, modifications in the social organization of the classroom, changing roles of teachers and students and changing attitudes towards learning are all central mediating factors of computers' effects [20], [119]. Conversely, in a two-year ethnographic study of the use of Quill in two sixth-grade classrooms, Michaels found that the social organization of the classroom shaped the nature of the tool and its uses [88], [89]. For example, there was no use of the tools for collaborative writing or writing across the curriculum in these classrooms because those were not a part of the teacher's normal classroom management and teaching style. The tool was used simply as a typewriter to type final manuscripts after they had been handwritten and hand edited on paper. Similarly, although the tool had been designed to encourage student-initiated writing, it was not used in that manner because the teachers assigned all the writing tasks and topics.

The "Talking TextWriter" is a computer program that combines word processing with synthesized speech, enabling young learners to use the computer as a tool to acquire beginning literacy. Teresa Rosegrant, designer of the program, has studied the use of Talking TextWriter with children of various ages, under a variety of conditions [115], [116], [117], [118]. She found significant differences in the quantity of writing, strategies for revising, and quality of writing products for children using the speech output in comparison with children not using speech output. She has studied the specific ways in which children at different developmental levels use particular features of the tool, and has documented the characteristics of adult behavior in the tool environment which are most conducive to the children's development of literacy.

The use of word processors in the writing and language arts classroom is generating new phenomena and issues. For example, there is more collaboration between teacher and student and among students. This creates new questions about how to evaluate individual students' work. Another effect is the loss of an audit trail through one's revisions unless one consciously stores each version during the process. Thus teachers and students lose opportunity to reflect upon the process of revision.

Tools Support Social Studies Inquiry and Problem Solving

Data bases, data analysis tools, and simulations are highly appropriate tools in support of the social studies ([66], [21], [24], [43], [46], [58], [60], [61], [67], [102]). They are used for a wide variety of instructional purposes in a wide variety of ways--inquiry, problem solving, analyzing and predicting trends and relationships, forming and testing hypotheses, developing inductive and deductive logic, applying research skills, applying social participation skills, and providing access to specialized collections of information.

Dozens of curriculum materials packages supporting data base use in the social studies have been published, both by commercial publishers and regional education consortia. A few formal studies have been made of the ways in which such materials are useful in meeting the social studies objectives (e.g. White [133], [134]; Parker [107]). A difficulty in performing controlled studies of this type is the lack of suitable instruments and methods for measuring either process or learning outcomes.

The Bill of Rights unit of *The U.S. Constitution Then and Now* package [51] provides students with an opportunity to play active, creative roles in analyzing important contemporary cases involving the Bill of Rights. Assuming roles as investigators, students work in much the same way attorneys and paralegals work, using data bases to locate and organize precedent court cases that will support their position on cases. Instructional objectives for the unit include research skills such as analyzing documents for relevance to an issue or position, and social skills such as participating in persuading and negotiating in resolution of conflicts.

Tools Support Math Concept Formation and Problem Solving

Spreadsheet programs are sometimes used in applied mathematics classes to enable students to focus on higher level aspects of problem solving while the computer performs the computations. Programs to graph functions enable teachers and students to view many and varied examples of functions and their graphic representations, thus aiding intuition concerning the functions. Computer algebra programs such as muMATH [42] can perform complex algebraic manipulations usually taught by rote memorization of algorithms. Such programs are believed to free students to concentrate on more important conceptual issues, such as problem formation, problem solving, and evaluation of problem solutions.

A mathematical microworld is a system composed of objects, relationships among objects, and operations that transform objects and relationships (Thompson [125]). An example is the "Geometric Supposer." It allows students to conjecture about geometric relationships, to construct those relationships, examine them in a graphic environment, and reach conclusions supported by the tool-generated evidence. In his research with students using the mathematical microworld MOTIONS, Thompson [125] found that such tools can effect significant improvements in students' mathematical understandings and abilities [124], [126]. He finds that most students benefit from using mathematical microworlds by exploring them systematically, such as by controlling variables while manipulating others, and by using visual feedback to debug faulty understandings whenever they lead to unexpected results. However, he finds a significant minority of students are effectively blocked from taking advantage of the tool if they fail to understand the initial concepts depicted by a microworld, do not know how to explore ideas, and/or do not know how to generalize from examples

Modelling Tools and Environments Support Complex Problem Solving

Using modelling environments such as STELLA, curriculum developers can set up models for students to investigate, or students can create their own models to represent problems. Such models of complex, real-world physical, biological and social systems involve solving coupled, nonlinear differential equations. Before students know the formalism of calculus or the algebra of derivatives and integrals, they explore the underlying concepts of change and accumulation which are at the heart of calculus. Bob Tinker at TERC and Nancy Roberts at Lesley College are developing prototype software and learning materials for students in grades 7-12 to work with complex systems [128].

Microcomputer Based Laboratories Aid in Scientific Inquiry

Microcomputer-based laboratories (MBL) directly link temperature, motion, sound or light-sensitive probes to a microcomputer. Data measuring changes in such parameters as pendulum oscillations or light intensity are collected and instantaneously displayed on the computer screen [127]. Students control how the data will be represented (such as a graph of changing parameter vs time, or a pictorial representation like a thermometer), length of experiment, and quantitative ranges over which data are represented. Such tools enable students to test and retest hypotheses as the experiments progress, rather than performing all data analysis long after the experiment is completed. Since the computer is performing the clerical work of recording and storing the data, students have more time to focus on the purpose of the experiment and the analysis and interpretation of the data.

Studies of Graphing Skills

The use of graphs as a central means of communication in MBL raises important questions, such as: "What are the prerequisite skills students need to accurately interpret the graphs? Does MBL facilitate the acquisition of these skills? " Several studies have been made of the impact of MBL on children's graphing skills (Mokros [90], [91], [92], [93]; Nachmias & Linn [97]; Linn [8], [73], [74], [76]; Layman & Kirkpatrick [72]; Stein *et al.* [121]; Mokros & Tinker [94]; Barclay [4]). The studies show that students do develop ability to interpret graphs in an MBL environment, and also that they are susceptible to a wide variety of misconceptions. One problem is their uncritical assumption that a graph accurately represents the physical phenomenon being studied, such as temperature, rather than being a function also of the equipment and software used (e.g. a probe may not be calibrated correctly). Other studies are beginning to look more closely at the particular features of the MBL interface that affect students' cognitive development. In a study with 93 high school physics students from rural schools in Florida, Brasell [14], [15] found that one class session with MBL produced significantly greater ability to interpret distance-time graphs (but not velocity-time graphs) than control groups. Most of the improvement was directly attributable to the real-time nature of the graphing output in MBL. This was demonstrated with a control group of students who used the MBL interface with a brief (seconds) delay inserted into the graphing interface. This control group did no better on the learning task than students who did not use MBL at all.

Microworlds Support Science Inquiry

Several versions of microworlds for exploring Newtonian mechanics have been developed. For example, the "ThinkerTools" microworld developed at BBN Laboratories is designed to help children understand the nature of scientific laws and how they are established. The children induce and evaluate the laws of Newtonian mechanics through experiments with visual representations of objects that respond to various forces. [135]. Researchers are just beginning to identify the features of such microworlds that contribute to their effectiveness—such factors as multiple representations fo concepts and information, dynamic feedback, learner control and motivation.

Networking Tools Facilitate Science Collaboration

Earth Lab is a prototype classroom microcomputer networking system for the study of geography in upper elementary and middle schools. Earth Lab is designed for collaborative science. Geography units engage students in scientific activities such as compiling and analyzing databases and keeping electronic laboratory journals [101].

Kids Network is a science learning environment in which students share data they have collected and analyzed, such as data on the acidity of their local rainfall, with students in their own class, other classes, and other schools. Bob Tinker expects 1,000 schools from several countries to be involved in Kids Network in the next 2 or 3 years [128].

Data-Based Laboratories Support Science Inquiry

Data bases serve as flexible laboratories for developing and testing hypotheses and analyzing data from laboratory experiments (McLeod & Hunter [84], [85], [86]). *Weather and Climate Laboratory* [87] is a data-based laboratory in which students develop and test hypotheses concerning relationships among variables that affect weather. For the Weather Unit, data on fifteen variables such as wind direction, dew point, sky cover and temperature are provided on disk for each three-hour reporting period for each day in two months, July and January, for one location. This fine-grained data base enables students to analyze local and short term changes in weather and develop concepts such as the relationship between precipitation and changes in wind direction. Some of the data are in spreadsheet files, so that students can insert their own formulas and manipulate the data mathematically. In the Climate unit, students use data from the National Weather Service on 77 locations throughout the U.S., selected to represent different latitude, longitude and proximity to landforms and bodies of water which affect climate. They work with 13 variables for each month for twelve months for each location. Concept papers and lab report templates for each experiment, are in word processing files. By the end of the unit, students are able to hypothesize what combination of variables will result in precipitation, and test their hypothesis against the data.

Tools Aid in Development of Information Processing Skills

The need for students to develop skills and understandings in the handling of information is becoming more widely appreciated [1], [67], [11]. Data bases and electronic encyclopedias are used to help students develop and apply information-handling skills. Instructional materials are available to teach such skills (e.g., [68], [69]). Krendl & Fredin [71] compared students' use of print encyclopedias versus electronic encyclopedias to prepare term papers. The researchers examined whether students' thinking processes, as represented in their term papers, differed as a function of the delivery medium. Papers written by students who used the electronic encyclopedia to gather information scored significantly higher on measures of both horizontal (broad and shallow) and vertical (narrow and deep) knowledge. Use of the electronic encyclopedia facilitated searching across content areas, using key words to move around in the system to conduct a horizontal search. The print encyclopedia, on the other hand, was expected to facilitate focusing on a narrow topic and researching it in depth, reading one complete entry after another. Results suggested that students' high levels of interest and motivation resulting from use of the electronic encyclopedia yielded significantly higher levels on both types of knowledge. Another study [133] investigated whether the manipulation of a computerized file management program contributed to positive gains in students' ability to locate, gather, organize and evaluate information. Participants included 14 secondary social studies teachers and 665 students. Computer-using students scored significantly higher on the information processing scale than the control students.

However, if students do not have the prerequisite information handling skills they cannot take advantage of the tools. In an observational study [41], eighth-graders used a videotext system to conduct electronic database searches. They found that students had limited skills. They mastered only the simplest videotext commands and did not improve their rate of error over time; had difficulty understanding the concepts and vocabulary of the electronic menus; accessed only a small quantity of electronic text compared to their use of available print materials; were able to search by keyword; and required constant adult assistance to log on to the videotext service, learn command operations and store text on disk files. Several other observational studies of students' problems in using tools have been published (e.g. Hawkins & Kurland [55]; Freeman ***et al.*** [50]; Hawkins, Char & Freeman [54]; Lockheed ***et al.*** [78]; Michaels [89]).

Tools Support Collaborative Learning

Small group decision making and problem solving can be strongly supported by tools such as data bases and simulations. The student groups are given a problem or decision making assignment. They use the data base or simulation to help them refine their problem or create an hypothesis, test the hypothesis or decision, analyze the data, interpret the findings or results of the decision, and communicate these results orally or in writing. For example, the data base may be designed to support role-playing in an historical simulation. In the Constitutional Convention unit of *The U. S. Constitution Then and Now* [51] students work as members of state delegations to the Convention of 1787. Three data files (Delegates, States, and Voting) and 76 word processing files stored on AppleWorks support the teacher and students in several ways. They:

- provide information.
- provide tools for students to use in retrieving, analyzing and organizing information.
- provide tools for students to use in communicating ideas and information among each other.
- provide structure and guidance to help organize the role-playing process.
- enable the teacher to tailor the activities to his or her own objectives and resources
- provide a means for students to create meaningful products from their work

The small group collaboration mode of tool use enables students to develop and apply skills in small-group participation which are important goals in the social studies. The data files, if properly structured, provide not only data but also structure for the small group processes.

Metacognitive Skills: Computer Tools for Learning Through Reflection

A major value in solving problems occurs when students reflect on how they actually solved the problem and how the strategies they used might be improved. Computer-based tools can capture the processes by which a learner carries out a complex problem solving task. This "audit trail" can become a useful object of study for students who are trying to learn how to improve their performance on a task or how to monitor their own learning processes (Collins & Brown [29]). The tool software built into Algebraland [17] and the Geometry Tutor [2] provide two examples. As the student takes steps in solving the problem (solving an algebraic expression for a particular variable or constructing a geometry proof), the tool creates a diagram showing the solution states the student reached and the operators the student used to reach those states. At any given point in the process the learner can observe her prior efforts, reflect upon them, and attempt a different strategy. Without such aids, the student might flounder endlessly, without learning from the experience.

More Tools

There are many more tools than those mentioned here. For example, school use of online data bases is increasing rapidly ([3], [62], [79], [106], [109]). New tools are being developed as the technology becomes more powerful (e.g., [56], [139]). Although an amazing amount of work has been done in the past few years in providing such tools to children in school, we have barely begun to understand the implications of these intellectual assistants for the development and self-actualization of children.

NEEDS FOR RESEARCH AND DEVELOPMENT

In this section we address the needs for research, development and evaluation in the following strategic areas:

- research on learning and teaching
- curriculum reassessment and reform
- evaluation methods and instruments
- teacher preparation and enhancement
- development and application of technologies
- social issues

Research on Learning and Teaching Using the Tools

We need deeper understanding and better theories about how children think and learn in complex problem-solving situations. There are hundreds of questions, at many levels. Some examples: What are the effects of different knowledge representations on children's understanding? How do different tool interface features affect children with different learning styles? In what ways does the ability to manipulate information in the form of pictures, graphs, sound and motion as well as text affect the learner's ability to conceptualize in a given subject area? What misconceptions are introduced by the knowledge representation employed? What characteristics of the tools constrain or inhibit children's cognitive, social or affective development? At what stage in learning a skill or subject is a particular tool most helpful? What roles does or can the tool play in enhancing or diminishing or shaping the child's metacognitive abilities? What characteristics of the tools influence affective factors in problem solving? In what ways can the tools facilitate or inhibit collaborative learning? How much ambiguity in problem specification is useful at what stage in a child's development? What prerequisite skills are needed before the learner can productively take advantage of the tool? What misconceptions are introduced or reinforced by the design of the tool or its interface? What characteristics of a tool inhibit or enhance a learner's ability to transfer the skills to new situations? What instructional methods are effective in helping students effect the transfer and application of skills to new problem areas? What level and kind of adult functioning is eventually attained by children who begin at a young age using powerful tools to address complex problem solving tasks and are consistently provided such environments throughout their schooling?

At least some of this research needs to be performed within the context of the tools and learning environments actually available to and used in schools, rather than being focussed solely on advanced technologies available only in research laboratories. A major need is for systematic, long-term ethnographic studies of classroom events and interactions.

Curriculum Reassessment

Attempts to integrate tools into the curriculum will have limited benefits unless curricula are reassessed and modified, both in goals and in classroom practice [64]. What skills, knowledge and values are needed in the information age, both within and across disciplines? Some efforts are being made in this direction, but much more needs to be done. The state of California, for example, has integrated critical thinking skills into each grade level curriculum in social studies. James Fey at the University of Maryland has been examining the implications of numeric, symbolic and graphic information processing capabilities on what students can and should learn in school mathematics [48]. Expert study groups in all disciplines have for years been calling for increased curriculum emphases on *processes:* formulating problems; locating, evaluating, analyzing, interpreting and synthesizing information; estimating reasonableness of solutions; iterating and refining; communicating problems, methods, findings and interpretations to others. (e.g. [13], [98], [99], [100]). Yet in practice the curriculum is still highly content oriented with oversimplified problems, lower level skills required of students, performance of rote procedures, and little room for creativity or individuality. Other school constraints such as 45-minute class periods mitigate against work on solving problems on meaningful complexity.

Evaluation Methods

Hand in hand with the need for curriculum reassessment is the need for new methods of 1) evaluating students' progress and diagnosing their individual needs for learning; and 2) evaluating the effects of technology based tools and their characteristics on the learning process.

Students, teachers, school administrators and school districts are now evaluated on the basis of student scores on standardized tests. Given the influence of these test scores on school curricula, textbooks, classroom activities, teacher evaluation, student placement, and the like, we must ask whether the tasks students are asked to perform on these tests are representative of the kinds of tasks, skills and attitudes students need to be developing in the information age. The model of norm-based, paper and pencil tests with multiple choice answers is being called into question and needs to be reassessed in light of emerging process-oriented goals in education. Alternative methods and instruments need to be explored and tried out. Work has been done on so-called adaptive interactive testing, but these are mostly traditional paper and pencil tests administered online. Other models may be possible. For example, can mechanisms be built into the design of the learner's tools that will assist the learner in assessing his own progress or assist the teacher in assessing the learner's needs? Can computer-based assessment tools be built which provide the learner with a set of problems along with problem solving tools and data to manipulate, and observe the learner's processes of solution? Researchers at the Educational Testing Service (ETS) are beginning to explore these possibilities.

Teacher Preparation and Enhancement

Increasing attention is being paid to needs, issues and problems surrounding the teaching profession in general and the relationship of teachers to technology in particular. Recommendations for action were made as part of the Conference on Technology and Teacher Education at U.C. Berkeley last year [75]. Visionary proposals for the reform of the teaching profession and the preparation of teachers have been put forth (e.g. Carnegie Forum [25]; Holmes Group [59]). A variety of approaches need to be tried, and teachers must be empowered to take a lead role in creating and assessing these approaches [130]. High technology learning environments in which students are working creatively and collaboratively on complex problems using powerful tools and large amounts of information, are diverse places which no single model of "the teacher role" will fit. Many different models of classroom environment and teacher roles need to be experimented with and be observed by teacher educators, teachers and novice teachers. Creative and flexible approaches to state regulation of the teaching profession need to be studied and tried. Collaborative work among teachers within a school needs to be encouraged. For example, library/media specialists can work closely with a history teacher to plan and conduct learning activities in which students use online data bases as part of their research projects [62].

Development and Application of New Technologies

In this time of rapid technological advances, there is always a discrepancy between the capabilities of the machines already installed in schools and the capabilities of the machines available in the general economy. Educational software developers and publishers who depend on school purchases of their products are forced by the economic system to confine their development efforts to the installed base of machine capabilities in the schools. If new technology applications to elementary and secondary education are to be ready by the time they are needed and wanted by the schools, work must be initiated now. Nearly all the advanced technology work in instruction today is funded by and for the military. The National Science Foundation Science Education Directorate has a small (approximately $5 million) budget for the Advanced Technology program. This program funds a small number of highly innovative and risky projects which are expected to have a five to ten year time frame before any payoff is expected, and multi-year funding is a critical component. The program focuses on educational application of emerging technologies such as laser disc and computer-generated imagery in four strategic areas: knowledge based systems and intelligent tutors; computational tools and symbolic manipulation systems; authoring systems; and programming and problem solving. More support is needed for the design, development and evaluation of promising new technology tool applications specifically aimed at elementary and secondary education. These projects should have relatively long-term (5 to 10 years) development cycles. They should not be constrained by the installed base of equipment in the schools, but rather take advantage of the most advanced technology needed to try out the ideas.

One development area that is just beginning to be explored is the creation of interactive coaches or advisors that operate on top of or in conjunction with the tool [112]. These could assist students in planning, problem definition, critical thinking, reflecting on their strategies, and other skills which are difficult for the teacher to assist in at all times, without taking away from the learner control inherent in the tool philosophy. Such an approach combines the expertise of those involved in research on intelligent tutoring systems, cognitive task analysis, mental modelling, and tool-building.

Another promising area is educational applications of large multimedia data bases. Much experimentation needs to be done with such systems in the areas of interface, pedagogy, browsing and retrieval capabilities, classroom management, curriculum integration, relationship to textbooks. Work especially needs to be done in the area of knowledge representation and appropriate learner interface with various knowledge domains. This becomes more critical as large scale data bases on optical media become more affordable. Sophisticated tools such as hypertext software are becoming more readily available for use with large data bases, and open new opportunities for design of appropriate systems for students (e.g. Yankelovich [139]; McClintock [82]).

A third promising area for development of educational applications is the use of object-oriented languages such as SmallTalk and logic-oriented languages such as Prolog, as tools of the learner to create and experiment with their own microworlds (e.g. Ennals [43], [44], [45]; di Sessa [36], [37], [38], [39]; di Sessa & Abelson [40]).

Another important technology application is in the design of tools to support collaborative work. Local area networks and telecommunication networks can provide the hardware base for collaboration, but research and development is needed on software supports for collaboration among students, teachers, and "the outside world."

Social Issues

Myriad ethical and values issues are arising faster than our culture has been able to address them -- issues involving intellectual property rights, privacy, honesty in communication, equity of access to information, to list just a few. As the tools of the information age begin to pervade the learning environment of children, it becomes imperative that they be taught the ethical, legal, moral and social aspects of the uses of the tools. Much work is needed in understanding social issues involved in use of electronic technology; deriving from that understanding the implications for what students need to be learning and valuing; and integrating these requirements into teacher education, curriculum materials, and features of technology tools.

SUMMARY

A wide variety of computer-based tools are available to assist learners in all subject areas, and interest in the use of these tools in elementary and secondary schools in the U.S. is growing rapidly. Research on learning and teaching using such tools has barely begun. Many new and sometimes subtle questions are beginning to be asked about how children think and learn in the information-rich, complex problem solving environments made possible through the technology. Attempts to integrate use of the tools into the curriculum and current classroom practice will have limited benefit unless curricula are reassessed and modified to take into account the kinds of skills and knowledge needed in the information age. Hand in hand with the need for curriculum reassessment is the need for new methods of evaluating students' progress, diagnosing their individual needs for learning, and evaluating the effects of technology-based learning environments on the learning and teaching processes. High technology learning environments in which students are working creatively and collaboratively on complex problems using powerful tools and large amounts of information, are diverse places which no single model of "the teacher role" will fit. Many different models of classroom environment and teacher roles need to be experimented with, and teachers must be empowered to take a lead role in creating and assessing these approaches. Long term projects are needed to design, develop and evaluate promising new technology applications for elementary and secondary education.

REFERENCES

(1) AFIPS. *Computers, Information, and Related Technologies*; Report of the AFIPS Secondary Education Project Working Committee (Forthcoming) *Education & Computers International Journal.* Amsterdam: North Holland Press.

(2) Anderson, J.R., Boyle, C.R. & Reiser, B.J. (1985) Cognitive principles in the design of computer tutors. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*. Boulder, Co: University of Colorado, 2-9.

(3) Aversa, E.S. & Mancall, J.C. (1987) Online users in schools: A status report. *Online,* May, 15-18.

(4) Barclay, W.L. (1986) Graphing misconceptions and possible remedies using microcomputer-based labs. In *Proceedings of the National Educational Computer Conference (NECC).*

(5) Baty, A. *et al.* (1986) A review of database software for science: A MicroSIFT Quarterly Report. Northwest Regional Educational Laboratory.

(6) Becker, Henry J. (September 1986) *Instructional Uses of School Computers: Reports from the 1985 National Survey.* Baltimore, MD: The Johns Hopkins University. Center for Social Organization of Schools, 2.

(7) Becker, H.J. (November 1986) *Instructional Uses of School Computers: Reports from the 1985 National Survey.* Baltimore, MD: The Johns Hopkins University. Center for Social Organization of Schools, 3.

(8) Berger, C.F. (1982) Attainment of skill in using science processes. I. Instrumentation, methodology and analysis. *Journal of Research in Science Teaching,* 19, 249-260

(9) Berger, C.F. & Pintrich, P.R. (in press) Attainment

of skill in using science processes: II. Grade and task effects. *Journal of Research in Science Teaching*.

(10) Berger, C.F., Pintrich, P.R., & Stemmer, P.M. (in press) Cognitive consequences of student estimation on linear and logarithmic scales. *Journal of Research in Science Teaching*.

(11) Bezold, C. & Olson, R. L. (1986), *The Information Millennium: Alternative Futures*. Washington, DC: Information Industry Association.

(12) Bowen, B.L. & David, J. (1985) *Teachers can't wait - grant program*. Cupertino,CA: Apple Education Foundation, Apple Education Affairs, Formative Evaluation.

(13) Boyer, E.L. (1983) *High School: A Report on Secondary Education in America*. The Carnegie Foundation for the Advancement of Teaching. New York: Harper & Row.

(14) Brasell, H.M. (1987a) The effect of real-time laboratory graphing on learning graphic representations of distance and velocity. *Journal of Research in Science Teaching*, (24) 4, 385-395.

(15) Brasell, H. (1987b) *The role of microcomputer-based laboratories in learning to make graphs of distance and velocity*. Paper presented to the annual meeting of the American Education Research Association at Washington, D.C.

(16) Brown, A. L., Bransford, J. D., Ferrara, R. A., & Campione, J.C. (1983) Learning, remembering and understanding. In Flavell and Markham (Eds.), *Carmichael's Handbook of Child Psychology, 3*, New York, NY: John Wiley and Sons.

(17) Brown, J.S. (1985) Process versus product: A perspective on tools for communal and informal electronic learning. *Journal of Educational Computing Research, 1* (2), 179-201.

(18) Brown, J.S., Moran, T.P., & Williams, M.D. (1982) *The Semantics of Procedures: A Cognitive Basis for Training Procedural Skills for Complex System Maintenance*. CIS working paper, Intelligent Systems Laboratory, Xerox Palo Alto Research Center, California.

(19) Bruce, B., Michaels, Sa. & Watson-Gregeo, K. (1985) How Computers can change the writing process. *Language Arts*. Feb.

(20) Bruce, B.C., & Rubin, A.D. (in press) What we're learning with QUILL. In M.L. Kamil & R.C. Leslie (Eds.) *Perspectives on Computers and Instruction for Reading and Writing*. Rochester, NY: The National Reading Conference.

(21) Budin, H.R., Kendall, D, & Lengel, J. (1986) *Using Computers in the Social Studies*. New York, NY: Teachers College Press.

(22) Burton, R.R. (1987) The environment module of intelligent tutoring systems. In Conference Minutes, *Research Planning Forum for Intelligent Tutoring Systems*.

(23) California State Department of Education. (1986) Technology in the curriculum series: History-Social Science; Language Arts; Science; Mathematics: *Technology in Curriculum Resource Guides*, February

(24) Canning T. R. & Brown, S. W. (Eds.) (1986) *The Information Age Classroom, Using the Computer as a Tool*. Irvine, CA: Franklin, Beedle and Associates.

(25) Task Force on Teaching as a Profession (Ed.) (1986) *A Nation Prepared: Teachers for the 21st Century*. Washington, D.C: Carnegie Forum on Education.

(26) Cohen, C., Craig, B., & Strucker, J. (1986) The variable impact of computers on children. Paper presented at the *Penn Ethnography and Education Research Forum Center for Urban Ethnography*. University of Pennsylvania.

(27) Cohen, M. & Riel, M. (1986) *Computer Networks: Creating Real Audiences for Student's Writing* (Report 15) La Jolla, CA: University of California, Interactive Technology Laboratory.

(28) Collins, A. (1986) Teaching reading and writing with personal computers. In J. Orasanu (Ed.), *A Decade of Reading Research: Implications for practice*. Hillsdale, NJ: Erlbaum.

(29) Collins, A. & Brown, J.S. (in press) The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning Issues for Intelligent Tutoring Systems*. New York, NY: Springer.

(30) Collins, A. & Gentner, D. (1980) A framework for a cognitive theory of writing. In L. W. Gregg & E. Steinberg (Eds.), *Cognitive Processes in Writing: An Interdisciplinary Approach*. Hillsdale, NY: Erlbaum.

(31) Collins, A. & Stevens, A. L. (1982) Goals and strategies of inquiry teachers. In R. Glaser (Ed.), *Advances in Instructional Psychology II*. Hillsdale, NJ: Erlbaum.

(32) Copeland, W.D. (1985) Teaching students to "do" history: The teacher and the computer in partnership. *The History Teacher, 2* (2), 189-197.

(33) Daiute, C. (1985) *Writing and Computers*. Reading, MA: Addison-Wesley.

(34) Dervin, B. (1983) Information as a user construct: The relevance of perceived information needs to synthesis and interpretation. In S.A. Ward & L. J. Reed (Eds.) *Knowledge Structure and Use: Implications for Synthesis and Interpretation*. Philadelphia, PA: Temple University Press. 153-183.

(35) Deucher, L. (1986) AppleWorks: Data disks, strong nations, and powerful learners. *Apple Education News*. Jul-Sep.

(36) di Sessa, A. (1985) A principled design for an integrated computational environment. *Human-Computer Interaction, 1* (1), 1-47.

(37) di Sessa, A. (1986) Artificial worlds and real experience. *Instructional Science, 14* (3/4), 207-227.

(38) di Sessa, A. (1986) Notes on the future of programming: Breaking the utility barrier. In D. A. Norman & S.W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Lawrence Erlbaum.

(39) di Sessa, A. (in press) The third revolution in computers in education. *Journal of Research in Science Teaching*.

(40) di Sessa, A., & Abelson, H. (1986) Boxer: A reconstructible computational medium. *Communications of the ACM, 29* (9), 859-868.

(41) Eastman, S.T. & Agostino, D. (1984) Commanding the computer: Student's uses and understanding of videotex technical paper. Presented at the annual meeting of the Speech Communication Association, Chicago.

(42) Educational Technology Center. (1987) muMATH-80 symbolic mathematics system for the Apple II family of computers. Unpublished manuscript, University of South Carolina, College of Education, Columbia.

(43) Ennals, J.R. (Ed.) (1986) *Information Technology & Education: The Changing School*. New York,NY: John Wiley & Sons, Inc.

(44) Ennals, J.R. (1985) *Artificial Intelligence: applications to logical reasoning and historical research*. New York, NY: John Wiley & Sons.

(45) Ennals, J.R. (1984) *Beginning Micro-PROLOG*. (2nd Revised Edition) Ellis Horwood and Heinemann & New York, NY: Harper and Row.

(46) ERIC (1982) Using data bases in social studies. (ERIC Fact Sheet No. 5) Boulder, Co.: Clearinghouse for Social Studies/Social Science Education.

(47) Eylon, B. & Reif, F. (1984) Effects of knowledge organization on task performance. *Cognition and Instruction*, (1), 5-44.

(48) Fey, J.T. (Ed.) (1984) *Computing and Mathematics*. Reston, VA: National Council of Teachers of Mathematics.

(49) Fisher, G. (1986) Enhancing incremental learning processes with knowledge-based systems. In H. Mandl & A. Lesgold (Eds.) *Learning Issues for Intelligent Tutoring Systems*. New York, NY: Springer-Verlag.

(50) Freeman, C., Hawkins, J., & Char, C. (1984) *Information Management Tools for Classrooms: Exploring Database Management Systems* (Technical Report No. 28) New York, NY: Bank Street College of Education, Center for Children and Technology.

(51) Furlong, M., Hunter, B., & Finkel, L. (1987) *The Constitution Then and Now: AppleWorks Files for U.S. History and Government Classes*. New York, NY: Scholastic, Inc.

(52) Gerrard, L. (1987) *Writing at Century's End*. New York,NY: Random House.

(53) Hannah, L. (1986) Social Studies, spreadsheets and the quality of life. In T.R.Canning & S.W.Brown (Eds.) *The Information Age Classroom, Using the Computer as a Tool*. Irvine, CA: Franklin, Beedle and Associates.

(54) Hawkins, J., Char, C., & Freeman, C. (1984) Software tools in the classroom. Unpublished manuscript, Center for Children and Technology. New York, NY: Bank Street College of Education.

(55) Hawkins, J., & Kurland, M. (1984) Analysis of software tools. Unpublished manuscript, Center for Children and Technology. New York, NY: Bank Street College of Education.

(56) Hawkins, J., & Pea, R. (FY 1985 - FY 1986) Inquire: Software tools for science education. *Research supported by the National Science Foundation*. New York: Bank Street College of Education.

(57) Hawkins, J. & Sheingold, K. (1986) The beginning of a story: Computers and the organization of learning in classrooms. In J.A. Culbertson & L.L. Cunningham (Eds.), *Eighty-fifth Yearbook of the National Society for the Study of Education*. Chicago, IL: University of Chicago Press.

(58) Hodges, J. O. (1985) Using database management to achieve social studies objectives. *Virginia Resolves, 27*, 6-14.

(59) The Holmes Group (Ed.) (1986) *Tomorrow's Teachers: A Report of The Holmes Group*. East Lansing, MI: The Holmes Group, Inc.

(60) Hunter, B. (1983b) Powerful tools for your social studies classroom. *Classroom Computer Learning*, Oct, 50-57.

(61) Hunter, B. (1983c) Social studies education in the information society. *Social Education, 47* (5), 321-324.

(62) Hunter, B. (in press) *Online Searching in the Curriculum*. Santa Barbara: ABC-Clio Inc..

(63) Hunter, B. (1985) Problem solving with data bases. *The Computing Teacher*, 12, 20-27. Reprinted in T. R. Canning, & S.W. Brown, (Eds.), (1986) *The Information Age Classroom, Using the Computer as a Tool*. Irvine, CA: Franklin, Beedle and Associates.

(64) Hunter, B. (1986a) What is fundamental in an information age? A focus on curriculum. *Education & Computing. 3*, 63-73.

(65) Hunter, B. (1987a) Knowledge-creative learning with data bases. *Social Education*, Jan, 38-43.

(66) Hunter, B. (1987b) Data Base applications in elementary and secondary school social studies education: state of the art. *In Proceedings for the International Conference on Data Bases in the Humanities and Social Sciences.* Auburn University at Montgomery, Alabama.

(67) Hunter, B. (in press) Information literacy: Skills for the information age. *Education & Computers,* North Holland Press.

(68) Hunter, B. & Devan, S. (1986) *Working with Scholastic pfs: File*. New York,NY: Scholastic, Inc.

(69) Hunter, B., Devan, S., Furlong, M. & McLeod, R. (1987) *Data Bases for Bank Street Filer*. New York, NY: Scholastic, Inc.

(70) Johnson, R.T., Johnson, D.W., & Stanne, M.B. (1985) Effects of cooperative, competitive, and individualistic goal structures on computer-assisted instruction. *Journal of Educational Psychology*, *77* (6), 668-677.

(71) Krendl, K.A. & Fredin, E.S. (1985-1986) The effects of instructional design characteristics: An examination of two communication systems. *Journal of Educational Technology Systems, 14* (1), 75-86.

(72) Layman, J.W. & Kirkpatrick, D. (1987) *A Laboratory Practicum to Test Students' Microcomputer Based Laboratory Skills*. Unpublished manuscript. College Park, MD: University of Maryland, Science Teaching Center.

(73) Linn, M.A. (1987) *Learning More --with Computers as Lab Partners*. Berkeley, CA: University of California, Lawrence Hall of Science, 1-5.

(74) Linn, M.C. (1986a) Computer as lab partner. *Teaching Thinking and Problem Solving, 18* (3), Lawrence Erlbaum Associates: Hillsdale, N.J.

(75) Linn, M. C. (Ed.) (1986) Education and the challenge of technology. *Proceedings of a Conference on Technology and Teacher Education*. Monterey, California.

(76) Linn, M., Layman, J. & Nachmias, R. (1986) Cognitive consequences of microcomputer-based laboratories: Graphing skills development. *Journal of Contemporary Educational Psychology.* Manuscript submitted.

(77) Lockheed, M.E. & Mandinach, E. B. (1986) Information technology and education: trends in educational computing: decreasing interest and the changing focus on instruction. *Educational Researcher*, May

(78) Lockheed, M., Gulovsen, J., & Morrison, D. (1985) Student use of applications software. (Technical Report.) Cambridge, MA: Harvard Graduate School of Education, Educational Technology Center.

(79) Lodish, E. (1987) Classmate to 2100 Dialog at Montgomery Blair. *Online*, May, 27-31.

(80) Malone, T.W. (1981) Toward a theory of intrinsically motivating instruction. *Cognitive Science, 4*, 333-369.

(81) Marcus, S. (1987) Personal communication.

(82) McClintock, R. (1986) Into the starting gate: on computing and the curriculum. (C C & T Technical Reports #4.) New York,NY: Teacher's College, Columbia University,

(83) McKenzie, D.L. & Padilla, M.J. (1984) Effects of laboratory activities and written simulations on the acquisition of graphing skills by eighth grade students. Paper presented at the meeting of the National Association for Research in Science Teaching, New Orleans, Louisiana, April.

(84) McLeod, R. & Hunter, B. (1985a) *Scholastic pfs: Curriculum Data Bases for Life Science*. New York, NY: Scholastic, Inc.

(85) McLeod, R. & Hunter, B. (1985b) *Scholastic pfs: Curriculum Data Bases for Physical Science*. New York, NY: Scholastic, Inc.

(86) McLeod, R.& Hunter, B. (1987) The data base in the laboratory. *Science and Children*, Jan 28-30, 155.

(87) McLeod, R., Hunter, B., & Finkel, L. (1987) *Weather and Climate Lab: AppleWorks Data Files for Science Curriculum*. New York, NY: Scholastic, Inc.

(88) Michaels, S. (1986) *The Computer as a Dependent Variable*. unpublished manuscript. Cambridge, MA: Harvard Graduate School of Education.

(89) Michaels, S. (1985) Classroom processes and the learning of text editing commands. *Quarterly Newsletter of the Laboratory of Comparative Human Cognition. 7*, (3) 70-79.

(90) Mokros, J.R. (1985) *The Impact of Microcomputer-Based Labs* (Technical Report 85-2) Cambridge MA: Technical Education Research Centers.

(91) Mokros, J.R. (1986a) *The impact of microcomputer-based science labs on children's graphing skills.* Paper presented at the annual meeting of the National Association for Research in Science Teaching, San Francisco, CA.

(92) Mokros, J.R. (1986 April) *The impact of MBL on children's use of symbol systems.* Paper presented at the meeting of the American Educational Research Association, San Francisco, CA.

(93) Mokros, J.R. (1987) *What Children are Learning About Graphs.* Unpublished manuscript.

(94) Mokros, J.R. & Tinker, R.F. (1987) The impact of microcomputer-based science labs on children's ability to interpret graphs. *Journal of Research in Science Teaching, 24*(4), 369-383.

(95) Montgomery County Intermediate Unit (1986) *Integrating Databases into the Curriculum.* Developed by: Pennsylvania Department of Education. Bucks County Intermediate Unit, Montgomery County Intermediate Unit, Lehigh University RCRC and West Chester University RCRC.

(96) Morocco, C.C. (1986, July) *Microcomputers, Writing, and Learning Disabled Children: Field-Based Research.* Paper presented at Second Annual Special Education R & D Symposium, Washington, D.C.

(97) Nachmias, R. & Linn, M.C. (1987) Evaluations of science laboratory data: the role of computer-presented information. [Special Issue] *Journal of Research in Science Teaching, 24* (5), 491-506.

(98) National Council of Teachers of Mathematics (1984) *Computers in Mathematics Education.* Reston, Va: National Council of Teachers of Mathematics.

(99) National Science Board Commission on Precollege Education in Mathematics, Science and Technology. (1983) *Educating Americans for the 21st Century: A plan of action for improving mathematics, science and technology education for all American elementary and secondary students so that their achievement is the best in the world by 1995.* Washington, D.C.: National Science Foundation.

(100) National Science Foundation (1980) *What are the Needs in Precollege Science, Mathematics, and Social Science Education? Views from the Field.* Washington, D.C.: National Science Foundation.

(101) Newman, D. & Sheingold, K. (FY 1985 - FY 1986) Earth Lab: A classroom network for young scientists. *Research supported by the National Science Foundation.* New York: Bank Street College of Education.

(102) Nichol, J. & Dean, J. (1984) Pupils, computers and history teaching. In M. Yazdani (Ed.), *New Horizons in Educational Computing.* Ellis Horwood, Ltd.

(103) Nickerson, R. (1986) *Technology in Education in 2020: Thinking about the Not-Distant Future.* Background paper for Harvard Center for School Technology Conference. Cambridge, MA: Bolt Beranek, & Newman Inc.

(104) North Carolina Department of Public Instruction (1986), *County Hunters, Inc.*_ Data base activities for North Carolina Studies.

(105) Papert, S. (1980) *Mindstorms: Children, Computers, and Powerful Ideas.* New York,NY: Basic Books, Inc.

(106) Parisi, L. S. (In press) *Directory of Online Databases for High Schools.* Santa Barbara, CA: ABC-Clio.

(107) Parker, J. (1986) Tools for thought. *The Computing Teacher.* Oct, 21-23.

(108) Patterson, J.H., & Smith, M.S. (1986) The role of computers in higher-order thinking. In J.A. Culbertson & L.L. Cunningham (Eds.) *Microcomputers and Education, Eighty-fifth Yearbook of the National Society for the Study of Education.* Chicago, IL: University of Chicago Press.

(109) *Pennsylvania Online: A Curriculum Guide for School Library Media Centers* (1985) State Library of Pennsylvania, 333 Market St. Harrisburg PA, 17126-0333.

(110) Perkins, D. N. (1985) Information technology and education, The fingertip effect: How information-processing technology shapes thinking. *Educational Researcher.* Aug /Sep.

(111) Perkins, D. N. (1984) Creative thinking by design. *Educational Leadership, 42*,1, 8-24.

(112) Polson, M.C. & Richardson, J.J. (Eds.) (in press) *Intelligent Tutoring Systems.* Hillsdale, NJ: Erlbaum.

(113) Riel, M.M. (1985a) The computer chronicles newswire: A functional learning environment for acquiring literacy skills. *Journal of Educational Computing Research, 1* (3), 317-337.

(114) Riel, M.M. (1985b) Functional learning environments for writing. In H. Mehan et.al. *Computers in Classrooms: A Quais-Experiment in Guided Change* (Final Report to U.S. Department of Education, NIE).

(115) Rosegrant, T.J. (forthcoming) Using a microcomputer to assist children in their efforts to acquire beginning literacy. In J. Hoot (Ed.), *Word Processing and Early Literacy.* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

(116) Rosegrant, T.J. (1984) Fostering progress in literacy development: technology and social interaction. *Seminars in Speech and Language, 5*(1) 47-57.

(117) Rosegrant, T.J. (1986a) *It doesn't sound right: The role of speech output as a primary form of feedback for beginning text revision*. Paper presented at the annual meeting of the American Educational Research Association, San Francisco, April.

(118) Rosegrant, T.J. (1986b) *Talking Text Writer*. New York, NY: Scholastic, Inc.

(119) Rubin, A., Bruce, B., & the Quill Project (Sept 1983) *Quill: reading and writing with a microcomputer* (BBN Report No. 5410) Center for the Study of Reading, Cambridge, MA: Bold Beranek and Newman, Inc.

(120) Sopp, N.P. (1985) Do you really need a children's data base? *The Computing Teacher*. Nov, 43-45.

(121) Stein, J.S., Kirkpatrick, D. & Nachmias, R. (1986) *Computer as lab partner: students' subject-matter achievements*. Lawrence Hall of Science, University of California, Berkeley.

(122) Swett, S. (1985) All-American data files. *Teaching and Computers*. *2*, (8), 18-21.

(123) Talmis (1986) Annual Report.

(124) Thompson, P.W. (1985) A Piagetian approach to transformation geometry via microworlds. *Mathematics Teacher, 78*, 465-471.

(125) Thompson, P. W. (1987) Mathematical microworlds and intelligent computer-assisted instruction. In G. Kearsley (Ed.) *Artificial Intelligence & Instruction*. Reading, MA: Addison-Wesley Publ. Co.

(126) Thompson, P.W. & Dreyfus, T. (in press) Integers and algebra: Parallels in operations of thought. *Journal for Research in Mathematics Education*.

(127) Tinker, R.F. (1985) *Modeling and MBL: Software Tools for Science*. (Technical Report 85-4) Cambridge, MA: Technical Education Research Centers.

(128) Tinker, R. (1987) Personal communication. August 1.

(129) Tucker, M. (1985) From drill sergeant to intellectual assistant: Computers in the schools. *Carnegie Quarterly*, (3/4), 1-7.

(130) Tuthill, D., Seidel, S., & McClure, R. (1987) *An Ecological and School Specific Model of Student and Teacher Empowerment: The N.E.A.'S Mastery-In-Learning Project*. Paper presented to the annual convention American Educational Research Association, Washington, D.C.

(131) tum Suden, M. & Rowe, R. (1985) Teaching problem-solving strategies. In H. Mehan et.al. *Computers in Classrooms: A Quasi-Experiment in Guided Change* (Final Report to U.S. Department of Education, NIE).

(132) Weaver, D. (1986, May) Database software for Social Studies. *A MicroSIFT Quarterly Report*. Portland, OR: Northwest Regional Educational

(133) White, Charles S. (1987) Developing information-processing skills through structured activities with a computerized file-management program. *Journal of Educational Computing Research, 3* (3), 355-376.

(134) White, Charles S. (1986) Using data bases to develop information processing skills: assessing instructional effectiveness. *Proceedings of the 1986 Association for Educational Data Systems Annual Conference*, Eugene, Oregon Internationall Council on Computers for Education.

(135) White, Barbara (1988). Interactive technology and inquiry learning: what works. Paper presented at the annual meeting of the American Educational Research Association, New Orleans LA.

(136) Wisconsin Educational Media Association (1986) *Online Database Searching in the Curriculum*. Madison, WI.

(137) Wolf, D.P. (1985) Flexible test: Computer editing in the study of writing. *New Directions for Child Development, 28*, 6, 37-53.

(138) Wresch, W. (1987) *A Practical Guide to the Use of Computers in the English/Language Arts Classroom*. Englewood Cliffs, NJ: Prentice Hall.

(139) Yankelovich, N., Landow, G. P, & Cody, D. (1987) Creating hypermedia materials for English literature students. Institute for Research in Information and Scholarship, Brown University, Providence, RI. *SIGCUE Outlook*, (20).

THE ROLE OF HYPERTEXT IN TRANSFORMING INFORMATION INTO KNOWLEDGE

Christopher Dede
University of Houston—Clear Lake
2700 Bay Area Blvd.
Houston, TX 77058
(713) 488-9151

[This work was supported by funding from the Johnson Space Center, National Aeronautics and Space Administration.]

ABSTRACT

A new paradigm for how to use information technologies is emerging. Originally, computers were seen as number crunching machines; with time, their data processing capabilities were recognized. Now, the strengths of integrated computer and telecommunications devices for all forms of individual and group symbolic manipulation are being explored. One implication of this evolution is a shift in the goals and methods of education from an emphasis on acquiring information to a focus on transmitting knowledge.

Past generations of educational technology have used advances in hardware and software to increase the amount of data available, on the assumption that individual and institutional wisdom would thereby increase. In practice, however, high levels of data overwhelm students; they become unable to decide which information is important, to interconnect new information into existing knowledge, or to recognize overall patterns of meta-knowledge. Future generations of advanced instructional systems will use increases in power to deliver environmentally meaningful, contextually targeted, interconnected data ("knowledge"); and leading edge research is focusing on how computational devices can perform such a conversion of raw data.

Instructional systems can support other formats than text, and hypertext has considerable strengths as an emerging approach. Hypertext is a knowledge representation composed of nodes of information connected by links. This provides a framework for non-linear representations of knowledge.

Hypertext is a representation system well suited for conceptual exploration, retrieval, training, retention, group collaboration, customization, and revision. Several projects have focused on using a hypertext format to increase user mastery of the knowledge in a database. Based on that work, hypertext appears to be a powerful approach for knowledge transfer. The strengths of this representation are 1) immediate access to information scattered among various sources and 2) guided paths which sequence users' exposure to data and make interrelationships explicit. Cognitive principles of learning which support the concept of hypertext-based educational environments include active structural networks, schema theory, web learning, and generative processing.

Problems that may constrain the ultimate usefulness of hypertext are disorientation, cognitive overhead, combinatorial explosion, and collective communications dysfunctions. Creating joint human-machine systems involves the additional challenge of building cognitive partnerships between people and intelligent tools. Developing hypertext interfaces which facilitate knowledge creation, transfer, and utilization and which accustom people to non-linear representation formats will be crucial to further advances.

INTRODUCTION

A new paradigm for how to use information technologies is emerging. Originally, computers were seen as number crunching machines; with time, their data processing capabilities were recognized. Now, the strengths of integrated computer and telecommunications devices for all forms of individual and group symbolic manipulation are being explored. One implication of this evolution is a shift in the goals and methods of education from an emphasis on acquiring information to a focus on transmitting knowledge.

How do data, information, knowledge, and wisdom differ? Common usage of these terms conveys a sense of increasing complexity and utility, but a rigorous delineation of their subtle differences is much more difficult. As a simplified category system for the purposes of this paper, *data* will be defined as input gathered through the senses; and *information* as integrated data which denotes a significant change in the environment. (Anthropologist Gregory Bateson [1975] defined "information" as "any difference that makes a difference.")

Information is converted to *knowledge* by interconnecting it with known concepts and skills as part of achieving a goal. (Note that knowledge has an attribute of purpose, which implies the existence of an intelligent agent (human or computational) in transforming information into knowledge.) *Wisdom* is knowledge about knowledge ("meta-knowledge"). To illustrate, for a software programmer, the existence of a new primitive in the programming language would be data, an understanding of what new functions it adds to the language would be information, comprehension of how to use the primitive in coding would be knowledge, and mastery of when to use the primitive and of its effects on overall programming style would be wisdom.

Past generations of educational technology have used advances in hardware and software to increase the amount of data available, on the assumption that individual and institutional wisdom would thereby increase. In practice, however, high levels of data overwhelm students; they become unable to decide which information is important, to interconnect new information into existing knowledge, or to recognize overall patterns of meta-knowledge. Future generations of advanced instructional systems will use increases in power to deliver environmentally meaningful, contextually targeted, interconnected data ("knowledge"); and leading edge research is focusing on how computational devices can perform such a conversion of raw data.

For example, in business environments the concept of a Knowledge Base Management System (KBMS) is emerging. Such a system would include traditional management functions for large shared databases (i.e. data models, query languages, semantic integrity maintenance, storage/search structures, update optimization, concurrency, security, error recovery, and distribution of data and processes). Added artificial intelligence features might encompass co-existing knowledge representation schemes; effective control structures for deductive, plausible, and inductive reasoning; means for knowledge acquisition, refinement, and validation; explanation facilities; and dynamic human intervention [Brodie & Mylopoulos, 1986]. These systems will be adapted for educational usage (as were word processors, electronic databases, and spreadsheets), and instructional functionalities previously unavailable in classrooms will emerge.

THE EVOLUTION OF KNOWLEDGE BASED EDUCATIONAL DEVICES

Instructors in many fields and at all levels of education are increasingly finding that:

- communicating an overwhelming amount of interconnected information is essential for student achievement
- increasing the data pupils must absorb can overwhelm their ability to synthesize knowledge from this information
- implementing new approaches to managing knowledge in the current educational paradigm is very challenging

Conventional educational technology systems have failed to resolve these problems. Simultaneously, initiatives in artificial intelligence and cognitive science are creating new conceptions of knowledge creation, capture, transfer, and utilization. The trends and issues above require more functionality from an information source than passive storage of data; a technology-based "knowledge structure" which unobtrusively amplifies human cognition and memory is the next step in the evolution of educational devices.

Forces certain to impact on knowledge transfer include the rapid evolution of workstations (i.e. new capabilities, lower prices, standardization, the synthesis of computers and telecommunications) and of conventional large data base management systems (e.g. sophisticated use of templates, alternative options for mass digital storage, integration of multimedia information formats). Possible other developments during the next decade are 1) the widespread adoption of *hypertext* (non-linear storage of information) as an alternative knowledge representation system and 2) the emergence of advanced user interface capabilities (such as sophisticated environments for computer-supported cooperative work, "artificial realities" to facilitate design explorations, and intelligent tutoring systems) [Dede, 1986]. These innovations will drive a shift to an "information abundant" environment which has implications for the management policies, personnel training, and organizational culture of educational institutions.

THE IMPORTANCE OF THE KNOWLEDGE REPRESENTATION FORMAT

A fundamental aspect of any information system is its underlying knowledge representation. Symbols "re-present" reality, and the properties of any form of communication are shaped by the attributes of its symbolic substrate. For example, sometimes "a picture is worth a thousand words" because pictorial representational strengths are better suited to the needs of that particular knowledge transfer situation.

Representation formats are currently not well tailored to the needs of many task situations. As an example, consider software documentation. Historically, this type of information has been predominantly textual, consisting of annotations in the code itself coupled with descriptive written information.

Such a knowledge representation approach is limited in several ways. Descriptions based solely on text can be cumbersome compared to multimedia formats which allow easy incorporation of graphics, images, and animation. Also, text presents information in a sequential manner, but a person using software documentation frequently needs access to knowledge linked non-linearly (for example, examining the detailed exposition for an annotation in the code requires jumping around in a text-based documentation system). In addition, searches through textual material are constrained to finding strings of symbols in close linear proximity (e.g. locating all paragraphs in a document which contain the phrases "software engineering" *and* "Prolog" *and* "expert systems"). Such key word retrieval systems may locate less than twenty percent of the references relevant to a particular search [Blair & Maron, 1985].

THE CONCEPT OF HYPERTEXT

Instructional systems can support other formats than text, and hypertext has considerable strengths as an emerging approach. Hypertext is a knowledge representation composed of nodes of information connected by links. A *node* might contain text, data files, graphics, images, code, animation, or some mixture of these. Nodes can be arbitrarily large or small, but generally embody the equivalent of a few sentences to several pages of information; they can have multiple embedded links to other nodes.

Links are associative paths between nodes; they can be as simple as a connection between an origin and a destination or can have a variety of properties. Links allow the creation of a conceptual web with complex interdependencies among nodes. This provides a framework for non-linear representations of knowledge.

One way to conceptualize a simple hypertext system is to compare this architecture of nodes and links to long term human memory. Ideas are stored in the mind associationally; the word *apple* conjures up botanic, gustatory, computational, corporate, and theological dimensions. Imagine "apple" as a node, with multiple links spreading out to nodes describing these alternative meanings. Such a structure is one type of *semantic network* [Woods, 1975], in which links are determined by the relational meanings of the nodes' contents.

If an encyclopedia were organized on this basis, its knowledge would be accessible in more ways than the alphabetic sequential form currently used. Readers could browse by following links or could conduct searches which would include the link structure itself (e.g. find every paragraph which contains the words "apple" and "tree," but not "snake" ***and*** is linked to a paragraph containing the word "orchard"). Alternatively, if a visual representation of the link structure itself were available, one could navigate in the network of nodes by "flying" a route through the links or by "teleporting" from one node to another.

Shasha [1985] provides a more rigorous definition of such structures, describing hypertext as "text fragments embedded in a directed graph with labelled edges and instructions allowing users to traverse edges." He uses the concept of directed graphs in constructing a mathematically based *fragment theory* to optimize retrieval and presentation of knowledge. This work is directed toward developing a formal design strategy for electronic environments tailored to knowledge exploration.

USES FOR HYPERTEXT

A non-linear knowledge representation is superior to linear formats such as text for several types of applications. Researching a topic which is referenced in a variety of interrelated linear documents can be frustrating, since locating a part of the data required often provides little help for finding in other documents the remainder of the information needed. In contrast, a hypertext system allows the user to follow a web of connections in tracing knowledge scattered in multiple sources. (A metaphorical description of this attribute would be the first needle found in the haystack acting as a magnet which then collects the remaining needles.)

This property of hypertext also enhances the capability of documentation to serve as a source for training; a user unfamiliar with the knowledge space can be guided along prestructured paths. Currently, new workers often need substantial amounts of help from an expert human mentor. A major productivity gain would be possible if a hypertext documentation system could assume some of the responsibility for educating apprentices. In addition, the use of a non-linear representation which mimics human associational memory may give all users greater recall of the material in the knowledge base, since the network's conceptual structure mirrors their internal mental models of the information.

Another advantage of hypertext is that using text as a method of organizing thoughts about programming, authoring, design, or problem solving can be difficult, since a linear format does not facilitate collecting and integrating a variety of approaches to the task. Current outline processors are limited to sequencing ideas hierarchically but, when concepts can be interrelated in a more complex manner, the brainstorming and synthesis aspects of knowledge creation are easier. Hypertext as a representation enables both the capture of divergent mental models (as nodes) and their convergence into a coherent strategy (by linking these nodes into a semantic network).

Also, team collaboration on a project is empowered by a non-linear medium because annotations and suggested revisions can be readily incorporated into a document. For example, a chunk of code and related documentation could be circulated electronically to multiple reviewers; the comments of each could be iteratively attached, labeled, and linked until the final hyperversion captured the collective wisdom of the group. In contrast, collecting from team members opinions and amendments on a document in standard text format is very unwieldy.

Finally, hypertext formats support modularity of information, a vital strategy in large knowledge transfer systems. The same node of information can be referenced from multiple locations, minimizing duplication and overlap. This allows greater customization of documents (through tailoring the order and availability of segments), reduces the volume of archival material (since information used in many documents is stored in a single place), and facilitates revision (because only one node need be altered to create changes throughout the knowledge base).

Thus, hypertext is a representation system well suited for conceptual exploration, retrieval, training, retention, group collaboration, customization, and revision. The spectrum of knowledge functions a particular hypertext system supports will depend on what balance among these goal-directed activities is most important to users.

RANGE OF FEATURES

Describing hypertext applications in terms of the operations they enable can be quite complex. Conklin [1986] delineates a taxonomy of typical features which includes:

- support for hierarchical or non-hierarchical link structures
- allowing multiple versions of nodes or links
- support for paths (many links connected into a single permanent object)
- availability of functions which facilitate searches through the hyperdocument
- support for multiple concurrent users
- availability of sophisticated editing tools

Many of these characteristics would be essential for large scale instructional systems.

As "second generation" hypertext systems evolve, the range of features unique to non-linear representational approaches may expand considerably. For example, in current systems nodes and links are objects within hypertext; a next step would be the inclusion of higher order representational structures (e.g. subnetworks) as objects to be manipulated. Such *composite nodes* are already partially implemented in some hypertext systems.

Also, supporting different types of nodes is an important capability for hypertext systems, since part of their power is that a correspondence can be created between objects in the world and nodes in the hypertext knowledge base. By using a variety of node types, the user can model a real world phenomenon in a flexible network. Templates within typed nodes enhance the completeness of information entered and may aid in computational inference on node contents. Similarly, the availability of different link types allows the knowledge creator to move beyond a linear flow of information (in which concepts are either included or discarded) to a complex representation that gives the knowledge consumer a web of choices.

Focusing on nodes leads to analysis or reductionism, in which a complex phenomenon is understood by dividing it into smaller parts. In contrast, a link orientation promotes synthesis or systemic thinking, where the processes which connect components of an entity are the dominant determinants of its behavior. Hypertext as a representation supports both types of cognition: a structured network is analogous to an optical illusion, in which different images can be seen depending on which gestalt the observer superimposes on the information presented. This support for multiple mental models (recontextualization) is a strength of an associational format.

HYPERTEXT SYSTEMS FOR KNOWLEDGE TRANSFER

Several projects have focused on using a hypertext format to increase user mastery of the knowledge in a database. One such program, Intermedia, is being developed by the Institute for Research in Information and Scholarship (IRIS) at Brown University. This hypertext project is the latest in an evolutionary series of electronic documentation systems [Yankelovitch, Meyrowitz, & van Dam, 1985].

Intermedia enables browsing through multimedia information which is linked, cross-referenced, and annotated. System authors can access an integrated set of tools which include a text processor, a graphics editor, a timeline editor, a scanned-image viewer, and an application to view and manipulate three-dimensional models. Future plans include the development of segment, map, video, and animation editors, as well as access to CD-ROM data [Garrett, Smith, & Meyrowitz, 1986].

This hypertext system has been designed using object oriented programming to facilitate building the web structure [Meyrowitz, 1986]. Maps which present visual representations of the network are used to guide viewers through the system. Current research at IRIS is targeted to the development of sophisticated navigational systems, customizing the user interface, versioning, and "hot links" for propagating information through the web. Intermedia materials are used as a resource in several courses on the Brown campus, so the system is designed to handle a multi-user environment of workstations connected by local area networks.

A second group of hypertext knowledge transfer applications have been developed using HyperTIES (Hypertext based on The Interactive Encyclopedia System) at the Human-Computer Interaction Laboratory, University of Maryland. TIES was developed to facilitate user browsing through instructional databases with an underlying hypertext architecture. The system is designed to maximize ease of use for both readers and authors and has an explicit instructional model [Shneiderman & Morariu, 1986]. HyperTIES is being expanded to include an advanced browser (with string search, bookmarks, multiple windows, and user annotation) and sophisticated authoring capabilities. Videodisc and touchscreen support are also under development [Shneiderman, 1987a].

The Human-Computer Interaction Laboratory has conducted a series of rigorous empirical studies of various design alternatives for the user interface (e.g. mouse versus arrow keys, embedded versus explicit menus, alternative screen sizes). Since repeated research has shown that reading text from screens is thirty percent slower than from typewritten paper documents [Shneiderman, 1987b; Gould *et al*, 1987], such productivity studies are vital to the evolution of electronic documentation.

A prototype electronic encyclopedia with a hypertext substrate was partially developed at Atari [Weyer & Borning, 1985]. The four metaphors used to guide design of the knowledge base were a *model* (a representation of some knowledge), a *tour* (a particular path through some model), a *filter* (an intelligent interface which tailors a model to a particular user), and a *guide* (an intelligent electronic agent that selects tours and provides help). Object-oriented hypertext links and embedded simulations provided a richer representational structure than the typical linear encyclopedia format.

Another knowledge transfer hypertext project is Xanadu, an on-line proprietary system developed by Ted Nelson (who coined the terms "hypertext" and "hypertext" in the 1960s). Xanadu is designed to be generic as a hypertext application so that many different types of uses can be supported. The fundamental metaphor Nelson espouses is a unified, global literary environment [Nelson, 1987].

The fundamental node unit in Xanadu is the document, which may have windows (links) to any other document in the knowledge base. Versioning is supported, and sophisticated organizational and indexing strategies are under development. Nelson is devising a strategy for electronic accounting and distribution of royalties, since copyright issues are a major concern for electronic storage media.

An optical disc research prototype, Palenque, is being developed at the Bank Street College of Education [Wilson, 1987]. This knowledge transfer hypertext system uses a CD-ROM (Compact Disc--Read Only Memory) with General Electric's Digital Video Interactive (DVI) technology. Palenque is designed so that children and their families can browse a database about a Mayan temple; the information is organized both topically (as a museum with four theme rooms) and spatially (so that the user can "walk" through the temple). When supplemented by a structured network of graphics, sound, text, narration, slides, motion video, and simulation, *Virtual travel* is a powerful motivational strategy, as prior work with spatial data management attests [Lippman, 1984; Bolt, 1979; Mohl, 1982].

As an example of an application specialized to software development issues, A Computer Based Ada Training System (CBATS) with a hypertext format has recently been developed by Softech for the Pacific Missile Test Center in Point Mugu, California [Larson & Rienzo, 1987]. CBATS utilizes existing Ada documentation and training materials (e.g. Practitioner's Guide to Ada, Army Ada Training Curriculum) reconfigured into linked chunks which illustrate interrelationships. This system illustrates the strengths and limits of the rapid, low cost transfer of conventional information to a non-linear format.

A "second generation" hypertext system, HyperActivity, is currently being designed at the Microelectronics and Computer Technology Consortium (MCC). A precursor to HyperActivity was PlaneText, a proprietary Unix-based hypertext application [Gullichsen et al, 1986]. Also, an advanced three-dimensional graphics interface to semantic networks (SemNet) has been developed [Fairchild, Poltrock, & Furnas, 1987].

Frank Halasz, one of the architects of NoteCards, indicates that the HyperActivity project will implement the following features:

- composite, multi-media nodes
- dvanced search and query (using attributes and context)
- versions and histories
- sophisticated navigational displays
- support for collaborative work (e.g. contribution tracking)
- interface tailorability
- support for embedded inference engines

Such work on second generation hypertext systems is important because a space station electronic documentation system would require many of these features.

As the systems described above illustrate, hypertext is a powerful approach for knowledge transfer. The strengths of this representation are 1) immediate access to information scattered among various sources and 2) guided paths which sequence users' exposure to data and make interrelationships explicit. Cognitive principles of learning which support the concept of hypertext-based educational environments include active structural networks, schema theory, web learning, and generative processing [Jonassen, 1986].

OTHER HYPERTEXT SYSTEMS

This brief summary of illustrative research in non-linear knowledge representations has only scratched the surface. A detailed and technical overview of current hypertext systems is given in Conklin [1986a]; a summary for a more general audience, in Conklin [1987]. Commercial interest in hypertext is growing rapidly; three non-linear applications have appeared for Apple's Macintosh^{TM} and two for IBM PC^{TM} compatibles. Over the next several years, hypertext programs may proliferate as rapidly as expert system shells!

OPERATIONAL CHALLENGES

Limits typical of first generation hypertext systems (e.g. computational delays, deficiencies in visual representation of the web structure) will disappear over the next several years as second generation systems and more powerful workstations evolve. Deeper problems that may constrain the ultimate usefulness of hypertext are disorientation, cognitive overhead, combinatorial explosion, and collective communications dysfunctions.

Information which is organized in a complex manner poses a potential problem of user *disorientation*. In a linear medium, one can readily evaluate the extent to which a document's information has been traversed (how many pages read, how many left) and where a particular piece of data is located (chapter, section, paragraph). Large hyperdocuments may be more confusing. In a web of thousands--or millions--of nodes, how does one define a location in the network, establish a desired direction to move, or blaze a trail indicating those nodes already scanned? In a non-hierarchical structure, what type of system should be used to indicate where a piece of data has been stored? Research is underway to develop sophisticated visual-spatial interfaces to address these problems [Fairchild, 1984], but this problem is very challenging for networks with large numbers of nodes and links.

Even if a user familiar with a particular network experiences no disorientation, working in a hypertext knowledge base entails some extra *cognitive overhead*. If entering material, the author must think carefully about how to link the information being added to the web which already exists. At each node they encounter, users must choose which link to follow from multiple alternatives and must track their orientation in a complex multidimensional structure. The richness of a non-linear representation carries a risk of potential intellectual indigestion, loss of goal directedness, and cognitive entropy. Unless a hypertext system is designed carefully from a human factors perspective, increasing the size of the knowledge base may carry a cost of decreasing its usability.

The availability of multiple types of representations in a hypertext system compounds this problem. Representational alternatives allow users to tailor input to their individual cognitive styles and enable authors to choose a format well suited to the material being entered. However, coping with multiple formats adds to cognitive overload, and little is known about which representational ecologies are functional for different task situations.

Combinatorial explosion (being overwhelmed by alternative possibilities) is another type of potential limit. Consider the two extremes of interconnections among nodes: if all nodes are linked to each other, the network is meaningless; if each node is connected to only a preceding and a following node, the medium is linear. Hypertext formats are useful only when balanced between these extremes.

The number of links generated by adding an additional node to a network will vary depending on the type of knowledge being stored, the objectives of the documentation, and the sophistication of the user population. Suppose that many vital, subtle interrelationships exist in the network's material. Although some new nodes will simply annotate single existing nodes, a substantial proportion of nodes that are added may require multiple links. The larger the knowledge base, the more links will be needed to integrate the additional material (because more nodes which discuss related issues will already be present). In a worst case situation, a combinatorial explosion would take place, with each new node adding so many links that the difficulty of comprehending and maintaining the web exceeds the benefits which a non-linear format provides.

One strategy for solving this problem is to aggregate subnetworks into composite nodes which chunk material on a higher level of abstraction. Such an approach is being explored in second generation hypertext systems--but creating another dimension of hierarchy complicates the representational architecture and, unless implemented in a manner transparent to users, may increase disorientation and cognitive overload.

These potential limits are particularly acute in on-line, shared hypertext systems. The user of a collegial electronic knowledge base may find that, since last entering the system, familiar paths have changed and new material has appeared. Links that seem intuitively obvious to the author adding them may be puzzling to others. "Tower of Babel" *communication dysfunctions* are possible with a large knowledge base in which multiple users can alter the fundamental medium of interaction.

Possible problems of disorientation, cognitive overhead, combinatorial explosion, and collective communications dysfunctions reflect the intricacy of working with a knowledge base rather than a database. Knowledge is intrinsically complex, and transforming information to knowledge involves gaining a contextual, goal-directed understanding of the application domain. Utilizing an underlying representation based on hypertext will require more sophisticated skills--a new type of "literacy"--from personnel in every role which interacts with electronic documentation. However, knowledge is much more powerful in task performance than data or information, and the effort required to use a knowledge base for instruction may well be outweighed by gains in efficiency and effectiveness at every level of institutional operations.

EVALUATING THE USEFULNESS OF HYPERTEXT

How do hypertext and linear multimedia formats differ in their usefulness for knowledge representation? One approach to evaluating the strengths and weaknesses of each is to examine the underlying properties of these different formats. Any mechanically embodied intelligent knowledge representation system must have three elements [Brachman & Levesque, 1985].

First, a well-specified language is necessary to express the knowledge. This is not a trivial requirement, for the underlying properties of this language can be very influential in shaping the performance of the knowledge base. As an illustration, anthropologist Edward Hall [1973] describes how some American Indian languages treat colors as verbs (e.g. the sky *blues*), whereas in English colors are adjectives (the sky is *blue*). Modern physics has determined that colors are more like verbs than adjectives, and in fact many advanced optical concepts can be expressed (and perhaps discovered) more easily in languages where colors are active rather than passive attributes. Optimally, a knowledge representation language is tailored to the specific domain for which it will be used.

Second, a representation system must provide an inference mechanism for revealing to the user facts implicit in the knowledge. For example, if the knowledge base includes the propositions "Socrates is a man" and "all men are mortal," then the implicit fact "Socrates is mortal" should be readily accessible. Researchers in knowledge based systems are exploring alternative representational structures (i.e. rules, frames, schema) which enable such computational inferencing.

Third, work in artificial intelligence has determined that specific knowledge in a domain is very important to task performance (**"In the knowledge lies the power"**). As a case in point, consider the design of an expert system for software programming. One approach would be to construct a system which understood the general theory of Turing machines and had a powerful inferencing mechanism. This would obviously be inefficient compared to building a system with detailed knowledge about algorithms and language primitives likely to be useful. It is better to know than to be able to figure out [Szolovits, 1986].

Given this framework for evaluation, hypertext is theoretically superior to linear multimedia formats on several dimensions. The representation languages that hypertext can support are potentially more expressive because non-linear semantics (composite nodes) and syntax (attribute/value pairs, neighboring link structure, node and link types) are richer. Hypertext formats enable easier computational inference mechanisms because truth maintenance approaches can be used to propagate implications through the structured network. As discussed earlier, knowledge is contextual, goal-directed, and involves multiple perspectives; hypertext provides a regional semantic environment for a node, as well as supporting alternative mental models (analysis, synthesis).

Whether this potential is outweighed by the disadvantages of non-linear representations is a more complex question. Much depends on the type of material in the database and the goals of the person involved in knowledge creation, transfer, or utilization. For example, Shasha [1986] contrasts the activity of reading a short story collection with that of consulting a guidebook. A non-linear format would add little to the compendium of stories, which are intrinsically "chunked" and are not extensively cross-referenced. However, a hypertext guidebook that would allow searching for restaurants by location, price, or type of food would be most useful. In general, databases are good candidates for non-linear representation if they contain many pieces of information, each with multiple attributes, and if users need to see different groupings. These are typical characteristics of advanced instructional systems.

As discussed earlier, a hypertext format can be useful in situations involving conceptual exploration, retrieval, training, retention, collaboration, customization, and revision. A non-linear knowledge base which minimized problems of disorientation, cognitive overhead, combinatorial explosion, and collective communication dysfunctions could be very useful.

CONCLUSION

Linear formats are incapable of supporting the contextual sophistication needed in transforming information to knowledge. Of the representational alternatives, hypertext seems the most promising. Its strengths include:

- a similarity to internal human cognitive representations;
- the ability to represent environmental significance and goal-directed context in an associational manner
- an architecture which facilitates the use of many chunks of multi-attribute knowledge needed by different types of users;
- emerging robust implementations; and
- advantages in situations involving conceptual exploration, retrieval, training, retention, collaboration, customization, and revision.

Although very difficult, designing instructional systems which are optimal from the representational point of view is not a sufficient solution to the development of advanced knowledge structures. Creating joint human-machine systems involves the additional challenge of building cognitive partnerships between people and intelligent tools [Woods, 1986]. Developing hypertext interfaces which facilitate knowledge creation, transfer, and utilization and which accustom people to non-linear representation formats is crucial to further advances.

REFERENCES

Bateson, Gregory. (1975) *Steps to an Ecology of Mind*. New York: Ballantine.

Blair, D.C. and Maron, M.E. (1985) "An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System." *Communications of the ACM* **28**, 3 (March, 1985), 289-299.

Bolt, R.A. (1979) *Spatial Data Management*. Cambridge, MA: MIT Press.

Brachman, Ronald J. and Levesque, Hector J., Eds. (1985) *Readings in Knowledge Representation*. Los Altos, CA: Morgan Kaufmann.

Brodie, Michael L. and Mylopoulos, John (Eds.). (1986) *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*. New York: Springer-Verlag.

Conklin, Jeff. (1986a) A Survey of Hypertext. Austin, TX: MCC Technical Report STP-356-86, Microelectronics and Computer Technology Consortium.

Conklin, Jeff. (1987) "Hypertext: An Introduction and Survey". *Computer* **20**, 9, 17-41.

Dede, Christopher J. (1986) "A Review and Synthesis of Recent Research in Intelligent Computer Assisted Instruction." *International Journal of Man-Machine Studies* **24**, 4, 329-353.

Fairchild, Kim M. (1984) Computer Generated Synthesized Imagery. Minneapolis, MN: Computer Science Department, University of Minnesota.

Fairchild, Kim F., Poltrock, S.E., and Furnas, G.W. (1987) "SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases". R. Guindon (Ed.), *Cognitive Science and its Applications for Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Garrett, L. Nancy, Smith, Karen, and Meyrowitz, Norman. (1986) "Intermedia: Issues, Strategies, and Tactics in the Design of a Hypermedia Document System". *Proceedings of the 1986 Conference on Computer Supported Cooperative Work*. Austin, TX: Microelectronics and Computer Technology Corporation.

Gould, John D., Alfaro, Lizette, Finn, Rich, Haupt, Brian, Minuto, Angela, and Salaun, Josiane. (1987) "Why Reading was Slower from CRT Displays than Paper". *Proceedings of CHI+GI 1987*. New York: ACM.

Gullichsen, E., D'Souza, D., Lincoln, P., and Casey, T. (1986) The PlaneText Book. Austin, TX: MCC TR No. STP-333-86 (P), Microelectronics and Computer Technology Consortium.

Hall, Edward T. (1973) *The Silent Language*. New York: Doubleday.

Jonassen, David H. (1986) "Hypertext Principles for Text and Courseware Design". *Educational Psychologist* **21**, 4 (Fall, 1986), 269-292.

Larson, Anna C. and Rienzo, John C. (1987) Computer Based Ada Training System. Middletown, RI: Softech Technical Briefing Document.

Lippman, A. (1984) "Imaging and Interactivity". *Proceedings of the Fifteenth Joint Conference on Image Technology*. New York: ACM.

Meyrowitz, N. (1986) "Intermedia: The Architecture and Construction of an Object-Oriented Hypertext/Hypermedia System and Applications Framework". *Proceedings OOPSLA 1986*. New York: ACM.

Mohl, R.F. (1982) Cognitive Space in the Interactive Movie Map: An Investigation of Spatial Learning in Virtual Environments. Cambridge, MA: doctoral dissertation, Department of Architecture, MIT.

Nelson, Ted. (1987) *Literary Machines*. San Antonio, TX: Project Xanadu.

Shasha, Dennis. (1985) "NetBook: A Data Model to Support Knowledge Exploration." *Proceedings of the Very Large Data Base Conference*. New York: ACM.

Shasha, Dennis. (1986) "When Does Non-linear Text Help?" *Proceedings of the Expert Database Systems Conference*. New York: ACM.

Shneiderman, Ben, and Morariu, Janis. (1986) "Design and Research on The Interactive Encyclopedia System (TIES)". *Proceedings of the 29th Conference of the Association for the Development of Computer-Based Instructional Systems*. Washington, D.C.: ADCIS.

Shneiderman, Ben. (1987a) "User Interface Design and Evaluation for an Electronic Encyclopedia". *Proceedings of the Second International Conference on Human-Computer Interaction*. New York: North-Holland.

Shneiderman, Ben. (1987b) *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley.

Szolovits, Peter. (1986) "Knowledge-Based Systems: A Survey." Michael L. Brodie and John Mylopoulos (Eds.), *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database Technologies*. New York: Springer-Verlag.

Weyer, Stephen A. and Borning, Alan H. (1985) "A Prototype Electronic Encyclopedia". *ACM Transactions on Office Systems* 3, 1, 63-88.

Wilson, Kathleen S. (1987) The Palenque Optical Disc Prototype. New York: Technical Report No. 44, Bank Street College of Education.

Woods, William A. (1975) "What's in a Link: Foundations for Semantic Networks." Brachman, Ronald J. and Levesque, Hector J., Eds. *Readings in Knowledge Representation*. Los Altos, CA: Morgan Kaufmann.

Woods, David D. (1986) "Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems." *The AI Magazine* **6**, 4, 86-92.

Yankelovitch, N., Meyrowitz, N., and van Dam, A. (1985) "Reading and Writing the Electronic Book". *Computer* **18**, 10 (October, 1985), 15-30.

INFLUENCE OF ROLE AND FUNCTION UPON STRATEGIC PLANNING FOR EDUCATIONAL COMPUTING POLICY INITIATIVES

Brent Edward Wholeben
Senior Graduate Faculty
Department of Educational Leadership and Counseling
University of Texas at El Paso
El Paso, Texas 79968
(915-747-5300)

[This research was supported, in part, by a grant from the F.B. Cotton Trust, and the Regents of the University of Texas System. The information presented in this publication does not necessarily reflect the opinions of the granting agency. Endorsement by the University of Texas System should not be inferred.]

ABSTRACT

Professional educators responsible for policy development in the area of educational computing can expect to face several important tasks relative to policy component identification and priority conflict resolution. Classroom teachers' perceptions of policy priorities were significantly distinct from administrative and support services' perceptions, and the perceived needs of non-educators who represent the interests of the community. As artificial constraints were imposed, restricting the range of educational computing policy development, classroom teachers increased their priority divergence from those in the comparison groups. The linearity of difference between teachers and community perceptions, where administrative-oriented perceptions were routinely median, suggests a strong relationship between level of responsibility and proximity to the actual components of instruction, and the differential value of certain policy initiatives. When internal contrasts were effectuated, comparing instructional professionals with instructional-support professionals, a "role & function profile" could be statistically constructed in terms of perceived priorities for educational computing policy development.

INTRODUCTION

Current practices in many state and local education agencies, concerning the deployment of educational computing, are in violation of established legal rights of student and teachers (Wholeben, 1986a). Responding to the need to enact policy which ensured the professional deployment of computers in education (Wholeben, 1986b;1986c), the International Council for Computers in Education released in February 1987 the "Code of Ethical Conduct for Computer-Using Educators", the first comprehensive statement on ethics and equity in educational computing to address instructional, administrative, and support services' computing issues (ICCE, 1987). Leadership for policy development included international representatives from the United States, Canada, and Israel. Comprising 52 separate guidelines, this policy identified the major domains in educational computing: curriculum, access, privacy and confidentiality, teacher, student, community, organization, software, and hardware.

Concurrent investigations (Brooks, Wholeben, & Boswell, 1987) answered many global questions related to establishing a national policy on educational computing. Specific questions addressed parallel concerns: How might the professional educator react to such a comprehensive statement of behavioral objectives? Would this policy be accepted by the education profession as a reasonable guide to future action? Which issues would be considered sine qua non requirements? Which statements might be relatively unstable during periods of constraint (e.g., narrowed goal focus due to scant resources and fiscal exigency)? Since educational control is participatory in terms of external non-educator influences (e.g., school boards, trustees, regents, etc.), what differences might be contrasted between educators and non-educators in terms of overall strategy and/or motivation for implementing such a global policy on computing ethics and equity?

This present study considered two specific questions. First, how will the various components of policy concerning issues of ethics and equity in educational computing be assessed differentially based upon the concepts of role and function in the educational enterprise: supervision-peripheral (administrator, counselor, etc.), instruction-direct (teacher), and community-control (district board of education)? Second, what generic findings concerning human resource influences might assist the development of a professional profile for strategic computing policy planning initiatives?

BACKGROUND

This study redefines the earlier findings of the 1977 Stanford Research Institute study (Johnson, 1982) and the 1980 Rensselear Polytechnic Institute study (Parker, 1982) which emphasized six major issues: (1) objectives pertaining to unauthorized use; (2) rights to products, creativity, and actual property of software; (3) privacy and confidentiality; (4) organizational loyalty; (5) social effects of rapid informational access; and (6) responsibilities for influencing public opinion.

Policy guidelines authored in 1987 by the International Council for Computers in Education, addressing specific behavioral accountability on the part of computer-using educators, present multiple objectives for the nine domains of educational computing relevant to issues of ethics and equity: curriculum basis (8 objectives), system access (4< objectives), privacy and confidentiality (4 objectives), teacher as user (10 objectives), student as user (5 objectives), community relationships (5 objectives), organizational control (3 objectives), software resources (7 objectives), and hardware resources (6objectives). In this present investigation, each of 52 ICCE behavioral objectives were analyzed to differentiate contrasting priorities between individuals representing the key 'role & function' groupings in professional education:

supervisor, teacher, and community. The corollary stability of grouped policy priorities when artificial constraints are introduced was studied to understand changing human resource perceptions in education.

METHODOLOGY

Instrumentation

Three types of data were required: (1) general respondent demographics; (2) selection of initial policy priorities from among the total 52 components of the ICCE list; and (3) identification of which initial priorities change when environmental conditions present constraints upon initially perceived needs. The data collection instrument was self-disclosing by respondent, and contained a random ordering of the 52 ICCE policy elements, stated in declarative form as an objective of educational computing policy, and with individual schema for the respondent to report their responses to each item.

Subjects

This paper presents the results from an initially stratified, random sample of K-12 professional educators (n=793) from a three-state area (Texas, New Mexico, and Kansas). A non-educator, comparison group of equivalently-trained, business professionals was also constructed (n=56) to provide a control on educator v. non-educator policy interests in the development of policy governing the use of computers in schools. Based upon self-reported role and function, the initial educator sample was further delineated into two groups: supervision-peripheral (n=48), and instruction-direct (n=745).

Procedure

Respondents supplied three types of data in addition to completing the demographics section(s). First, to establish an initial priority ordering, each respondent selected 20 of the available 52 statements as perceived requirements for a comprehensive policy on computing ethics and equity. Second, to simulate an artificial environment of imposed general (viz. undefined) constraints, each respondent selected the most important 10 of these initial 20 components. Lastly, to differentiate relative worth among the top-10 choices, each respondent rank-ordered (from 1-10, without ties) these final 10 selections.

FINDINGS

Unconstrained

The peripheral group (viz., administrators and other professional educators concerned with instructional faciliation and support services rather than direct teaching) demonstrated significantly greater priority for policy concerning external influences (inter-agency planning, validating vendor services, and controlling gifts from hardware/software companies), when policy formulation was unconstrained. Significantly less priority was found in areas related predominately to the training of teachers in computer use (release time coupled with specific inservicing).

Contrasted with peripheral roles, the instructional group demanded significantly higher priority for policy related to teacher training (computer integration, specific use in classroom, and release time). Teachers also sought participation in the selection of appropriate hardware as well as specific investigation of various computer approaches dependent upon student learning style. Significantly less priority was demonstrated across a diversity of components, representing predominately the areas of: (a) evaluation (assessing results of computer use, validating vendor services, insuring proper use of system); (b) copyright (duplication of software and/or printed manuals); and (c) impact of external influences (promoting community awareness, controlling deployment of donor gifts, inter-agency planning).

The control group (community perception) in the unconstrained setting paralleled peripheral interests in the areas of lower priority for teacher training, and complementary release time for such training; and, furthermore, agreed that vendor services should be validated for accountability. However, while the peripheral group significantly disagreed with the instructional group only in the areas of teacher training and inter-agency planning, the control group was found to be widely opposed to teacher priorities. The control group demanded significantly greater priority for policy related to: (a) instruction for students in specific areas (vocational uses, ethical issues); (b) preservation of copyright (software and related-materials duplication); and (c) external influences (vendor validation, community awareness). Alternatively, low priority was found in areas related to: (a) teacher training (including release time); (b) participation in selection of equipment; and (c) individualization of computer use based upon personalized learning styles of students. By contrast, the control group demonstrated very low interest in developing specific criteria which would ultimately control the placement and use of hardware.

Constrained

When the range of policy options was restricted by the imposition of artificial constraints, respondents demonstrated continuing priority (sustained) for select policy components while identifying new significance (emergent) for other components. These changes in priority structures, influenced by constrained policy formulation, and contrasted by discrete classification of global responsibility, are summarized in Table 1.

The imposition of artificial constraints, restricting the diversity of policy-priority interests, encouraged the peripheral group to demonstrate greater priority for the areas: (a) evaluating software; (b) restricting the duplication of manuals; and (c) promoting community awareness of computer use. Furthermore, the peripheral group sought less priority for the differential application of computer-based instruction based upon learning styles, while maintaining initial low priorities for teacher training and corresponding release time.

The instructional group demonstrated little change in their initial interests when policy options were artificially restricted. Concomitantly, significantly greater priority emerged for parent involvement in planning, and a better understanding of curricular implications for computer use in the classroom. By contrast, the instructional group demanded less priority for specific instruction in computer ethics, and the planning of computer integration based upon future needs.

As reflected under artificial constraints, the control group displayed greater contrast with the instructional group, as compared to peripheral respondents. The control group increased priority in the areas of: (a) data confidentiality; (b) planning equity; and (c) strategic planning. Alternatively, less priority is established for the areas: (a) curricular implications of computer-use in the classroom; (b) evaluation of software quality; and (c) parent involvement in planning.

DISCUSSION

The comparison and contrast of differential policy priorities between identifiable demography groups, indigenous to the educational enterprise, provide a significant contribution to an understanding of perception, and resultant acceptance or rejection, of various policy initiatives related to educational computing. In this study, the linearity of difference between teachers and community perceptions, where administrative-oriented perceptions were routinely median, suggests a strong relationship between (a) level of responsibility and proximity to the actual components of instruction, and (b) the differential value of certain policy initiatives.

While peripheral respondents never significantly opposed the perceived policy initiatives of community (control group) respondents, when policy formulation was not constrained, these supervisory-oriented school personnel agreed with only 20 percent of demonstrated community needs when contrasted with teacher demands—predominately in the low-priority areas of teacher training and release time for training, and high-priority areas related to the importance of external influences. Similarly, peripheral respondents disagreed with approximately 27 percent of teacher perceptions when contrasted with community demands. On the other hand, teachers disagreed with 73 percent of all community initiatives related to educational computing policy development. Teachers demanded: (a) greater flexibility in reproducing computer-related materials regardless of copyright; (b) less concern with external influences; (c) general reliance upon the teacher as the validator of computer-use in the classroom rather than formal validation procedures; and (d) more attention given to the relationship between instructional computing and differential learning styles. These were opposed by community-perceived priorities.

As constraints were applied, peripheral respondents encountered their first, and only, opposition to community perception: acknowledging the importance of evaluating software quality. Alternatively, the peripheral group strengthened their agreement with community perception in terms of the importance of copyright protection and community awareness, and corresponding low priority for computer-use equity based upon learning style(s). Teachers increased their opposition to community policy demands by establishing new, high priority for parent involvement in planning, coupled with low priority for strategic planning based upon future needs rather than current demonstrative requirements.

Profile

When contrasts are restricted to internal role and function, comparing those who facilitate instruction (supervision-peripheral) with those who teach (instruction-direct), an internal "role and function profile" was statistically constructed related to priorities for educational computing policy development. This profile is summarized in Table 2.

Educational professionals who have supervisory/support responsibilities peripheral to instruction in the classroom will consistently acknowledge the importance of validating vendor services and promoting inter-agency strategic planning initiatives in computer-use development; and, furthermore, will consistently view the availability of release time for teacher training in computer use as low priority. Teachers, on the other hand, will view such differential priorities as directly opposed to their personal needs.

In an environment where policy initiatives are constrained or restricted, and priorities for a limited, select number of components becomes necessary, teachers will make new demands for teacher training in two areas: (1) specific instruction related to the use of computers in particular curricular environments; and (2) continual inservice related to new trends in society's deployment of computer technology; and, moreover, demand the incorporation of policy which guides the use of computers for instruction based upon differential learning styles of students in the classroom. Such initiatives will be contrary to supervisory interests. In addition, teachers will, under restricted conditions, discount policy which either prohibits the duplication of printed computer-materials or promotes community awareness of school-based computer utilization. These two elements, however, will be new, positive priorities for school administrators in the constrained setting.

CONCLUSION

Professional educators responsible for policy development in the area of educational computing can expect to face several important tasks relative to component identification and priority resolution. Perceptions of priority, contrasted by role and function, will differ, significantly, in terms of general goal focus as well as specific policy objective.

Administrators and other instructional support-services personnel will find themselves, literally, in-between the juxtaposed policy priorities of teachers and community citizenry related to educational computing. Attempting to support the demands of classroom teachers, these administrators will come into conflict with the community, realistically consolidated as a school board of trustees, parent group, etc.. Instructional personnel will find many of their perceived, basic necessities (e.g., personalized training, evaluative flexibility, and freedom from external influence) opposed by both school administration and the community. If policy development is restricted by budgetary constraints, cautiousness related to new-program development or expansion, or a generalized concern for deployment of technology for instruction, teachers will demand even greater flexibility and latitude, while administrators and the community will perceive the need for greater centralized authority and control.

The global policy initiative of the International Council for Computers in Education has provided an important first step for significantly improving the use of technology in elementary and secondary schools, and higher education, internationally. It remains the individual initiative of professional educators to study and adopt similar policy to satisfy their technological goals.

REFERENCES

Brooks, G. D., Wholeben, B. E., & Boswell, S. (1987). "Computer education in the United States of America: State policy on training, instruction, and control." *AFIPS Proceedings of the 1987 National Computer Conference* (Vol. 56). Reston, VA: American Federation of Information Processing Societies, pp. 205-213.

Johnson, D. G. (1982). "Educating toward ethical responsibility." In M. W. Hoffman, & J. M. Moore (Eds.), *Ethics and the management of computertechnology* (pp. 59-66). Cambridge, MA: Oelgeschlager, Gunn & Hann.

International Council for Computers in Education (1987, February). Code of ethical conduct for computer-using educators: An ICCE policy statement. *The Computing Teacher*, pp. 51-53.

Parker, D. B. (1982). "Ethical dilemmas in computer technology." In M. W. Hoffman, & J. M. Moore (Eds.), *Ethics and the management of computer technology* (pp. 49-56). Cambridge, MA: Oelgeschlager, Gunn & Hann.

Wholeben, B. E. (1986a). "Educational malpractice for computers in instruction: A perspective for legal advocacy." *AEDS Journal*, 19(2-3), 143-161.

Wholeben, B. E. (1986b). "Strategic planning for the design and development of management information system in education." *AFIPS Proceedings of the 1986 National Computer Conference* (Vol. 55). Reston, VA: American Federation of Information Processing Societies, pp. 185-192.

Wholeben, B. E. (1986c). "Evaluating existing administrative computing systems for future management information system development priorities." *Proceedings of the 1986 Conference of the Association for Educational Data Systems*. Rockville, MD: Computer Science Press, pp. 98-101.

Table 1

Constrained policy priorities by classification of global responsibility

	Peripheral	†(pct)
	more priority	
	(sustained)	
**	ACKN ETHIC OF QUAL H/W VENDORS (vendor validation)	29.8
**	PARTIC IN TAC/STRAT COMPU PLNG (inter-agency strategic planning)	27.7
	(emergent)	
**	EVAL S/W QUAL FOR CLASSRM USER (quality evaluation)	40.4
**	DISCOUR/REFUS PRINT MATER COPY (documentation duplication)	21.3
**	INC UNDERST OF EDUC COMPU GOAL (community awareness)	19.1
	less priority	
	(sustained)	
**	TM RELEAS FOR TRNG RE PROJ USE (release time for training)	38.3
**	TRNG APPROP TO CLSRM COMPU USE (pre-utilization training)	23.4
	(emergent)	
**	CURRIC MATER V DIFF LRNG STYLE (equity among learning styles)	25.5
	Instruction	†(pct)
	more priority	
	(sustained)	
**	TM RELEAS FOR TRNG RE PROJ USE (release time for training)	60.4
**	TCHR TRNG FOR INTEGRD COMP USE (teacher training for use and change)	45.2
**	TRNG APPROP TO CLSRM COMPU USE (pre-utilization training)	44.9
**	CURRIC MATER V DIFF LRNG STYLE (equity among learning styles)	42.9
**	TCH/ADM PARTIC IN EQUIP SELECT (participatory selection)	34.6
	(emergent)	
*	UNDERS COMPU DEV / CURR IMPLIC (intra-education literacy)	33.2
*	PRNT INVL IN COMP STRATEG PLNG (participatory planning)	14.5
	less priority	
	(sustained)	
**	CURR USE REFLCT FUTR STUD NEED (relevancy in vocational orientation)	30.3
*	PARTIC IN EVAL COMPU USE RESLT (participatory evaluation)	16.9
**	TCH SUBORDS DATA USE RESPONSIB (staff inservice development)	16.1
**	EXAM ISSUE OF COMPU USE ETHICS (situational assessment)	14.3
**	PARTIC IN TAC/STRAT COMPU PLNG (inter-agency strategic planning)	14.1
**	ACKN ETHIC OF QUAL H/W VENDORS (vendor validation)	14.0
**	DISCOUR/REFUS UNAUTHOR DUPLICT (program duplication)	8.9
**	DISCOUR/REFUS PRINT MATER COPY (documentation duplication)	8.7
**	INC UNDERST OF EDUC COMPU GOAL (community awareness)	8.6
	(emergent)	
*	TCHG STUD ETHICS OF COMPU USES (specific instruction in ethics)	17.7
*	EVAL SIMUL MODL FOR SOCIET DEC (ethics/equity bases in planning)	7.6
**	PLNG COMPU INTEG VS FUTR STRAT (strategic alternatives planning)	4.6

Table 1 (cont.)

	Control	†(pct)
	more priority	
	(sustained)	
**	CURR USE REFLCT FUTR STUD NEED (relevancy in vocational orientation)	51.9
**	TCH SUBORDS DATA USE RESPONSIB (staff inservice development)	37.0
**	EXAM ISSUE OF COMPU USE ETHICS (situational assessment)	33.3
*	TCHG STUD ETHICS OF COMPU USES (specific instruction in ethics)	29.6
*	PARTIC IN EVAL COMPU USE RESLT (participatory evaluation)	27.8
**	DISCOUR/REFUS UNAUTHOR DUPLICT (program duplication)	25.9
**	INC UNDERST OF EDUC COMPU GOAL (community awareness)	18.5
	(emergent)	
*	RESPCT OTH THRU COMPU DATA USE (data confidentiality)	27.8
*	EVAL SIMUL MODL FOR SOCIET DEC (ethics/equity bases in planning)	16.7
**	PLNG COMPU INTEG VS FUTR STRAT (strategic alternatives planning)	13.0
	less priority	
	(sustained)	
**	TM RELEAS FOR TRNG RE PROJ USE (release time for training)	27.8
**	CURRIC MATER V DIFF LRNG STYLE (equity among learning styles)	24.1
**	TCHR TRNG FOR INTEGRD COMP USE (teacher training for use and change)	22.2
**	TRNG APPROP TO CLSRM COMPU USE (pre-utilization training)	18.5
**	TCH/ADM PARTIC IN EQUIP SELECT (participatory selection)	16.7
	(emergent)	
*	UNDERS COMPU DEV / CURR IMPLIC (intra-education literacy)	20.4
**	EVAL S/W QUAL FOR CLASSRM USER (quality evaluation)	16.7
*	PRNT INVL IN COMP STRATEG PLNG (participatory planning)	5.6

†(pct) percent of response priority (**) p≤.05 (*) p≤.10

Table 2

Constrained policy priorities by classification of internal responsibility

	Peripheral	†(pct)
	more priority	
	(sustained)	
**	ACKN ETHIC OF QUAL H/W VENDORS (vendor validation)	29.8
**	PARTIC IN TAC/STRAT COMPU PLNG (inter-agency strategic planning)	27.7
	(emergent)	
**	DISCOUR/REFUS PRINT MATER COPY (documentation duplication)	21.3
**	INC UNDERST OF EDUC COMPU GOAL (community awareness)	19.1
	less priority	
	(sustained)	
**	TM RELEAS FOR TRNG RE PROJ USE (release time for training)	38.3
	(emergent)	
**	CURRIC MATER V DIFF LRNG STYLE (equity among learning styles)	25.5
**	TRNG APPROP TO CLSRM COMPU USE (pre-utilization training)	23.4
*	LRNG OPPORT RE COMPU SOC TRNDS (extra-education literacy)	19.1

	Instruction	†(pct)
	more priority	
	(sustained)	
**	TM RELEAS FOR TRNG RE PROJ USE (release time for training)	60.4
	(emergent)	
**	TRNG APPROP TO CLSRM COMPU USE (pre-utilization training)	44.9
**	CURRIC MATER V DIFF LRNG STYLE (equity among learning styles)	42.9
*	LRNG OPPORT RE COMPU SOC TRNDS (extra-education literacy)	31.6
	less priority	
	(sustained)	
**	PARTIC IN TAC/STRAT COMPU PLNG (inter-agency strategic planning)	14.1
**	ACKN ETHIC OF QUAL H/W VENDORS (vendor validation)	14.0
	(emergent)	
**	DISCOUR/REFUS PRINT MATER COPY (documentation duplication)	8.7
**	INC UNDERST OF EDUC COMPU GOAL (community awareness)	8.6

---

†(pct) percent of response priority (**) $p \leq .05$ (*) $p \leq .10$

SECONDARY DATA PROCESSING AND COMPUTER SCIENCE

Ernest Carey
Jerrie Crummett, Keborah Matheson
Frank J. Ferrari
Stephen Perelgut

ABSTRACT: Computer Science Contest for Hawaii High School Students

Dr. Ernest Carey, Coordinator of Academic Computing, Brigham Young University—Hawaii, Laie, HI

For the past six years, Brigham Young University—Hawaii has been hosting a computing contest for Hawaii high school students. The goals of the contest are to:

1. Provide an opportunity for Hawaii high school students to meet in an environment conducive to friendly competition in the area of computer science;
2. Promote a higher level of computer literacy;
3. Promote correct programming principles;
4. Motivate schools to teach Pascal, the language adopted for the Advanced Placement Exam in Computer Science; and
5. Provide leadership opportunities to BYU-H students in the Information Systems/Computer Science (ISCS) programs.

The contest features four main areas. Everyone takes a general quiz covering all areas of computer literacy and programming. A team of two traces of a program written in Pascal. A second team of two writes a Pascal program to solve a problem. One student gives a chalk-talk on a topic in computer science. Each high school enters a team of no less than three and no more than five students. For the past two years, some schools have elected to enter two teams, providing opportunities for additional students.

Six years ago, most programming courses in the high schools of Hawaii were using BASIC as the introductory language. Our contest was designed with that fact in mind and the programming and tracing problems were developed around a subset of BASIC commands common to the versions of BASIC then in use. We made it known to the participants that we would be slowly changing to an all Pascal format. We made the change in easy stages and by the fifth year all tracing and programming problems were provided in Pascal.

For the programming section of the contest we provide IBM PC, Macintosh, or Apple IIe systems and a variety of interpreters and compilers. If we do not have what the students are used to, they are invited to bring their own.

In this presentation, the contest categories will be explained and samples of the contest problems will be provided. Our successes and our areas of greatest challenge will be discussed.

ABSTRACT: Data Processing for the Visually Impaired

Jerrie Crummett, Instructor, DP/VI, Metro Tech, Oklahoma City, OK
Keborah Matheson, Instructional Aide, DP/VI, Metro Tech, Oklahoma City, OK

This course is an innovative program designed to equip the blind with the skills necessary to enter the work force. While we do practice diverse teaching techniques, we like to stress that the same curriculum is followed in our program as that in a standard data processing course. Admittedly the blind have to do things differently, but the end results are the same. Our goal is to produce employable individuals.

Today emphasis is being placed on economic growth. The non-working handicapped become an economic burden. The employed handicapped can only contribute to the economic growth of a community.

The objectives of this program include training the visually impaired on computers and familiarizing them with adaptive equipment. Our primary objective is to produce employable, visually impaired individuals, who possess the same levels of skills as sighted workers.

As well as introducing the visually impaired to the world of computers, we would like to familiarize society with the potentials of the visually impaired worker. The capabilities, knowledge, motivation, and technology all exist so that the blind can function as productive employees. We want to help society recognize this.

This presentation includes a computer actually talking to the presenters and audience. VERT, as we refer to the synthesizer, will introduce himself. VERT will also explain the capabilities and versatilities of voice synthesizers.

The explanation will include what VERT can do in the way of computer functions and software. Through VERT it will become clear to the audience how the blind can visualize a computer screen, and utilize their abilities to perform any data processing procedure.

VERT is talking software produced by Telesensory Systems Inc. of Mountainview, California. It has the ability to pronounce one character at a time or entire words. It possesses a dictionary of pronunciations that can be upgraded by the user. If the user wants a word pronounced a certain way. That pronunciation can be stored in VERT's memory. VERT can send special functions through the CONTROL or ALTERNATE keys. This allows any number of software packages to be compatible with VERT.

VERT can read one word at a time, one line at a time or the entire screen. It can read column blocks as well as horizontal lines, and windows can be set to avoid hearing the whole screen at times. Punctuation can be voiced or silent, and strange spellings can be spoken as words or spelled out to insure correct spelling.

Everything needed by the visually impaired user to accomplish data processing tasks is available through speech synthesizers and programs such as VERT.

This course teaches the same hardware and software taught in a standard data processing course. In addition, students learn about voice synthesizers and enlarging devices which open the world of computers to the blind. We work with IBM personal computers, and our software includes word processing, electronic spreadsheets, and database management systems. We also incorporate knowledge of three programming languages, and training on the IBM System 36 into the course.

Our presentation utilizes a personal computer and voice synthesizer. A BASIC program will allow the computer to interact with the presenter and audience. Through the voice synthesizer we will introduce the sighted to a phase of computers that need not be seen to be experienced.

The computer is able to voice to the user anything on the screen, or data entered through the keyboard. We will show examples of prompts, menus and other standard indicators through the synthesizer. The voice will sound somewhat foreign at first, but soon will become quite understandable.

ABSTRACT: Teleteaching Classes in Corrections Education

Frank J. Ferrari, Business Manager, Danville Area School
Steven P. Keifer, Teleteacher, Danville Area School District, Danville, PA

Danville, the seat of Montour County, is located in the scenic Susquehanna Valley of Central Pennsylvania, 100 miles west of the Pennsylvania-New Jersey border, 70 miles north of Harrisburg, and 50 miles southwest of Wilkes-Barre.

Health care and manufacturing industries are the primary employers in this otherwise agricultural region. Famous throughout the Commonwealth, Geisinger Medical Center, Danville's largest employer, serves residents of Central and Northeastern Pennsylvania with a 560-bed teaching hospital and associated clinic with nearly 360 staff and resident physicians.

Recently, the Danville Area School District was again selected as a pilot school for the Pennsylvania Teleteaching Project. This selection exemplifies the progressive, professional attitudes of the faculty, administration, and school board of the Danville District.

Teleteaching is an affordable educational delivery system that allows offering courses to students by linking students with a teleteacher, at a remote site. The system uses two regular telephone lines, microcomputers, modems, light pens, printers, and speakerphones. Special software has been developed for the Pennsylvania Teleteaching Project. Teleteaching is live team teaching between a master teacher and one or more associate teachers. To link three or more remote sites, a special equipment bridge is required for voice and data transmission.

Support from key administrators is essential to the success of Teleteaching.
Dr. William Opdenhoff, Superintendent of Danville Area School District, as well as William Mader, Chief of Corrections Education, Pennsylvania Department of Education, Harrisburg, have supported our project wholeheartedly. Without this support, the project would never have gotten off the ground.

The North Central Secure Treatment Unit (N.C.S.T.U.) is a maximum security facility for serious juvenile offenders. It was founded in January of 1979, due to the need for expanded secure care services in the Central Region, by the Department of Public Welfare (D.P.W.). North Central has a rated capacity of twenty-eight (28) students ranging in age from fourteen (14) to eighteen (18) years of age.

The North Central School Program is funded by the Department of Correctional Education which contracts with the Danville Area School District to supply an educational program to the students. At the present time, North Central has seven (7) Danville Area School District employees, six (6) full-time certified teachers and a school secretary.

All students placed at North Central attend school regularly. The educational diagnostic process includes achievement tests, as well as interviews and vocational interest and aptitude surveys. Students are placed in small classes according to academic ability.

The recently instituted Teleteaching Program presents a tremendous opportunity, not only for the North Central students, but quite possibly for the future direction of Corrections Education. Due to North Central's limited number of instructional staff and the wide range of our student's academic needs, Teleteaching affords the opportunity of offering curriculum areas not otherwise available to our small population.

Due to his failure in both school and the community, the North Central student is usually viewed as an unmotivated person who gives up easily and cannot endure adversity in any form. Teleteaching has presented an innovative and creative way to stimulate effort and improve the student's overall motivation and self awareness. The North Central Secure Treatment Unit administration is pleased to be involved with this new technology and looks forward to continued and expanded development in the future.

ABSTRACT: Teaching Programming in High School

Dr. Stephen Perelgut, Holt Software Associates, 203 College Street, Suite 305, Toronto, Ontario M5T 1P9 Canada

This presentation will discuss the results of a pilot project run by the University of Toronto that taught programming to 160 students in six different schools during the fall of 1987. Classes included students in grades 10, 11 and 12. The project was based on a new programming system developed at the university.

The project was a success and all 6 participating schools continue to use the software. This presentation will focus on how Turing was introduced to the schools, how students reacted, which system features students identified as difficult to master and where the pilot project results are leading.

The materials supplied with the project consisted of a student workbook and an integrated software package. The software, called the Turing Programming Environment, included an interpreter for the Turing programming language, an editor and built-in debugging and help facilities.

The Turing programming language was designed to be familiar to students. This means that the syntax looks somewhat like other common languages (especially Pascal and BASIC). This familiarity allows students to transfer skills easily across language boundaries.

Turing supports a concept we call levels of mastery. Individual features of the language can be taught in a modular fashion, allowing students to progress at their own pace. This also means fewer students feel overwhelmed by perceived complexities when they start to program. This modularity is reflected in the student workbook which is designed to help students write small programs immediately and to expand their knowledge of language features slowly, chapter by chapter.

Turing is concise and expressive, eliminating confusing syntactic elements. For example, the first program a student runs is usually one line long:

```
put "Hello, Stephen"
```

Thus a first-time programmer immediately finds the rewarding experience of controlling the computer.

Finally, Turing was designed to make programming fun. This is a hard goal to quantify but the pilot project results indicate it has been successfully met. Over three-quarters of the students participating in the pilot project indicated they wanted to continue to use the Turing System.

The interpreter for Turing was integrated with a full screen editor that allows students to create and modify source, run the programs, request help and debug the programs from a single environment. The students and teachers in the project were very pleased with the editing functions and the integration.

SOFTWARE TOOLS/AI CURRICULUM: COLLEGE

Carolyn Biediger
Jack W. Pope, Patricia Anderson, Sue Stokes, Paul Gordon
Norman Higgins
Lon Roberts

ABSTRACT: Hyped Psychology: Using Macintosh's HyperCard to Teach Junior College Psychology

Dr. Carolyn V. Biediger, Psychology Division,
Southwest Texas Junior College

In an attempt to deal more effectively with the range of student ability found in community colleges, I have written one unit of my General Psychology Course as an interactive tutorial using HyperCard on the Macintosh. HyperCard was used because its unprecedented power, speed of development, and branching ability make it ideal for software prototyping.

HyperCard enabled me to:

1. Present a structured outline of the topics;
2. Define any term or vocabulary word with one mouse click;
3. List all examples with a special example button under the outline heading;
4. Test as the material is presented to maintain attention and clarify concepts;
5. Use a variety of recall, recognition, and cued-recall test items;
6. Utilize an interactive method of concept teaching by presenting positive and negative instances, which I have found to be extremely powerful and effective;
7. Vary the presentation of reinforcers;
8. Intersperse short videotapes, readings, and written explanations in appropriate places;
9. Easily insert graphs, pictures, and diagrams;
10. Keep the presentation light, lively, and moving; neither predictable nor repetitious;
11. Use a branching format;
12. Present a report to the student in terms of objectives accomplished and objectives requiring more work;
13. Complete all of this in a matter of hours rather than hundreds of hours;
14. Keep a record of student responses for later evaluation; and
15. Edit presentation quickly in response to student feedback and performance.

The students are more willing to spend time at the computer than they are reading texts, and the branching tutorial lets the able student move on more quickly. The goal is to increase depth of processing and decrease student passivity in dealing with the material.

Our typical mainstream student is a Hispanic youth from a large family, income under 15,000 a year, and an ACT composite score of 12.3. About half of our student body comes from two of the poorest counties in the United States. They come to us with limited vocabularies, poor reading skills, and few study skills. Concept learning skills are lacking; the students are geared to memorization of facts.

For the evaluation of the project, the students are divided into three groups . The top group are students who will learn no matter what system is used, the bottom group will have great difficulty, and the two lower groups are the ones that hopefully will benefit from this intervention and remediation system.

ABSTRACT: New Tools for Student Training and Resource Management in the Computer Lab

Dr. Jack W. Pope, Professor, Mathematics/Computer Science; Director, Academic Computing
Dr. Patricia Anderson, Adjunct Professor, Education; User Services Consultant
Ms. Sue Stokes, Student Supervisor
Mr. Paul Gordon, Lab Supervisor
University of San Diego - Serra 188, Alcala Park, San Diego, CA

With the introduction of a new era of microcomputer hardware with expanded memory and enhanced graphics capability, new software for courseware and training modules are being developed which provide teachers with innovative and powerful authoring tools, enabling the teacher to compose interesting and effective instructional courseware without becoming bogged down in programmer-type languages.

One of the more recent such authoring tools is the HyperCard application developed by Bill Atkinson of Apple, INC. HyperCard is a extraordinary authoring tool which enables a user to create and organize learning modules, called "stacks" which the student can "browse" or "read through" in a number of different ways. In fact, with the appropriate controls, the reader can himself become a co-author, with the ability to add and create his or her own ideas in the "stack".

At the University of San Diego, Academic Computing is using HyperCard to develop a training and information base "stack" for its Work Study student lab assistants. The program, which leads users through a "tour" of all academic computing facilities, provides detailed information on all hardware and software used in its labs. Furthermore, in addition to its tutorial nature, the module can also be used as an information tool, using the "find" command to locate information on a given word or topic relevant to lab use. For example, the student can find information on "MINITAB", a VAX application, by using commands programmed into buttons for finding information on a key word or words.

The project presentation will include a progress report on the evaluation and effectiveness of this learning tool in the lab management environment, with particular emphasis on whether or not the tool is effective as an information resource as well as a training device for work study staff.

This project presentation is part of a larger interactive text group project at the University of San Diego, seeking to find new ways to enhance "reader" interaction with written material. The group recently received an equipment grant from Apple, Inc to aid in the program, and is using another interactive authoring program, called Storyspace, for other interactive text applications. While HyperCard is a commercial product of Apple, Inc., the application described is strictly a non-commercial product used on in the laboratory settings at USD.

ABSTRACT: Computer-Based Presentation Graphics

Norman Higgins, Professor, Educational Media and Computers, FMC/College of Education, Arizona State University, Tempe, AZ 85287

Graphics such as titles, tables, charts, graphs, and drawings are frequently used to support speaker presentations in education and business. This presentation describes a graduate-level course that is being developed at Arizona State University to teach students how to *compose*, *produce*, and *present* computer-based speaker support materials.

Students learn to compose graphic materials on Macintosh and IBM PC computers using a variety of drawing, painting, and graphing programs. They learn to produce slides and overhead transparencies using laser printers, thermal copiers, photocopy cameras, and slide duplicators. They also learn how to access outside production services to produce presentation materials. Finally, they learn how to present their graphic materials using slide, overhead and video projectors, liquid crystal displays, and computer displays.

The course objectives, the scope and sequence of course activities, and the equipment, programs, and materials used in the course are described in the presentation. Examples of student-produced graphic slides will be included in the presentation.

ABSTRACT: An AI Curriculum for Practicing Professionals

Dr. Lon Roberts, Director, Professional Education Resource Center, Association for Higher Education of North Texas, Dallas, TX 75252

The process of articulating new technical courses into a highly structured curriculum requires careful thought and detailed planning. Since technology changes at a quasi-exponential rate, academics must deal with this dilemma on an on-going basis. Within industry training and education programs the problem takes on still an added dimension of complexity, since the experience of the technical worker, as well as the needs of the employer, must also be taken into consideration.

Chaotic changes in technology (i.e., those that are rapid and pervasive) exacerbate this condition. The last 30 years have witnessed a handful of changes that might truly be considered "chaotic" in terms of their impact on technical professionals: the demise of the vacuum tube, advances in microelectronics, the transition from analog to digital systems, the mass proliferation of personal computers, computer aided design, to name a few. With the somewhat recent advent of commercially viable AI products, we are witnessing still another revolution in industry training programs.

This presentation describes the concept for an AI curriculum designed for practicing professionals. It provides an overview of the components of a "Professional Track in AI" and a review of the process and underlying philosophy that led to the evolution and design of this curriculum. It also examines the role of the Professional Education Resources Center (PERC) in executing the curriculum and sustaining the review and evolution process. Finally, it points out how the concept can serve as an effective model for bringing academe and industry into a mutually beneficial relationship.

ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: PROJECT DESCRIPTION AND BACKGROUND PAPERS I

Robert Taylor, *Chair*
Project Director
Teachers College, Columbia University
New York, NY

ABSTRACTS

One of the major concerns of computer-using school districts is how to train all of their teachers to be effective consumers of technology. Across the country millions of dollars are being spent retraining teachers to integrate computing into the curriculum. Yet with all this concern over the need for computer-using teachers, and over the cost of retraining current teachers, teachers, just now emerging from preservice programs are little more prepared than their existing colleagues.

The purpose of this project is to develop an interim set of guidelines for the addition of one solid course on the use of computers to existing preservice education programs. As a part of the development, papers were solicited from individuals to serve a background papers for the course development. These papers were reviewed at an invited conference held in January at Teachers College. The papers were then revised and published in the Spring '88 issue of SIGCUE Outlook. This is the first of two sessions at NECC to discuss the papers. Background papers to be discussed in this session are:

A Computers in Education Course for Preservice Teachers: An Historical Perspective

by Karen Billings and
David Moursund

Much of the early research and development work for the first computers was done on university campuses. Thus, it is not surprising to find that the history of computers in education closely parallels the overall development of the computer field.

However, it is only in the current decade that computers have become sufficiently inexpensive and available so that they are of major concern to precollege education. Most preservice teachers now learning about computers, and many of their instructors, lack an historical perspective of the computer in education field.

This paper traces the history of computers in education, focusing on long term trends. Major developments in computer hardware, computer software, and computer science are analyzed in terms of their impact or potential impact on education.

This paper also gives an overview of the current status of computers in education. This lays a foundation for asking and beginning to explore the question: "What are the unique new dimensions or capabilities that computers bring to education?"

One answer lies in Computer-Assisted Learning (CAL) which captures part of our increasing knowledge of both learning and teaching theory. Another answer lies in computer as tool to aid in problem solving. This includes generic tools such as word processor, spreadsheet, database, graphics, and telecommunications. It also includes more specialized tools, such as might be used in musical composition, or a sophisticated algebraic symbol manipulation package. All teachers need to develop a functional level of knowledge in the two areas CAL and computer-as-tool. Thus, they are core topics in a computer in education course.

A Model for Incorporating Teaching and Learning Theories into Preservice Computer Training

by Carl F. Berger and
Elizabeth A. Carlson

One of the critical elements in the design of preservice computer training experiences is the application of teaching and learning theories to computer use. Linking the fundamental concepts of various teaching and learning theories to the ways computers are used in schools of education will ensure that preservice teachers' use of computers take into account what is known about how humans learn.

This approach has two important components, the development of knowledge and attitudes that allow for successful computer use, and a comprehensive model of use; learning theory is essential to both. The model itself has three steps: identifying instructional goals and tasks, matching teaching and learning tasks to a computer based tool; and using these tolls to complete the task. Learning theories inform each aspect of this model of computer use, for both preservice instructors and students. Certain uses of computers support certain instructional tasks that follow from various theoretical approaches to teaching and learning. A rationale for computer use is created that is based on solving educational problems by combining principles of teaching and learning with the unique capabilities of the computer.

Teacher Training for Using Databases

by Howard Budin

The use of database managers and database software in schools, and enthusiasm for it among educators, has been growing rapidly. In spite of the lack of conclusive research, many have claimed that students' use of databases can enhance thinking skills, collaborative research, and global understanding, and can help transform the roles of teachers and students. This paper examines the obstacles, both technical and educational, that stand in the way of preventing the accomplishment of such goals, and suggests that preservice training, by having teachers experience the kind of database use that will effectively integrate desired goals into classroom instruction, can help overcome these obstacles.

Integrating Word Processing Tools with Writing Instruction: A Review of Research and Practice

by Amy Heebner

In the "writing with computers" segment of the preservice course, teachers-in-training should be exposed to key findings in writing research, in order to understand how computer tools may be integrated with the teaching of writing. Researchers confirm that writing is a process that consists of several distinct phases; however, individual writers work in unique patterns, not in a fixed sequence of events. Researchers also confirm the value of peer tutoring in word processing skills, and other kinds of social interactions in computer writing classrooms.

Research findings are supplemented by the experience of practicing teachers. The review of research and teaching suggests several essential ingredients for the preservice course segment, including: (1) hands-on experience with word processing tools; (2) exposure to writing process research; (3) first-person narrative as a starting point for student writing; (4) the development of student computer experts for peer tutoring. Other computer tools such as electronic mail and publishing programs have generated productive student writing activities, according to some researchers, and may also be integrated with the writing workshop.

Social Implications of Computers Ethical and Equity Issues

by Judith E. Jacobs

The introduction of computers into school settings has great potential for improving the instructional program of the elementary grades. Their presence, through can create situations in which individuals must make decisions that have ethical implications. In addition, deciding how the computers will be used, and by whom, raises concerns about the equitable use of computers.

This article provides classroom teachers with strategies for helping them address the ethical issues they will confront when their students have access to computers. Following a general discussion of the issues, activities are described that can be students. Examples of survey activities, discussions of moral dilemmas, role playing activities, and discussions of childrens' views on these issues are given.

Equitable use of computers occurs when all students are able to benefit, according to their individual needs and abilities, from using computers, regardless of the students race, sex, national origin, language or social class. A discussion of the participation of females and minority group members in computer activities is presented. This is followed by an analysis of factors that inhibit these groups' participation in computer activities and the presentation of recommendations for way of promoting computer equity.

The Role of Telecommunications in Preservice Training in Educational Computing: Content and Process

by Robert Spielvogel

Telecommunications in the form of electronic mail, computer conferences, online networks and information retrieval can play a dual role with in preservice training of educational computing. It is simultaneously a 1) necessary object of study and 2) a set of tools and experiences to promote effective learning in the other aspects of educational computing. Exposure and mastery of the basic elements of telecommunications provide a foundation for ongoing professional development of the teacher as well as a direct instructional focus for students. The paper identifies the key areas on which to focus preservice training, including; technical competencies, ability to relate the tools to specific curriculum areas, understanding of the ethical issues raised, and knowledge of how to provide relevant experiences given the varying degrees of access to these tools.

Preparing Future Teachers to Use Computers: A Course at the College of Charleston

by Francis C. Welch

Teacher preparation programs have the responsibility of preparing their undergraduate students to effectively use microcomputers to enhance the instructional process. This article describes the objectives and requirements for a course designed to teach undergraduate education majors at the College of Charleston to integrate microcomputer use into the curriculum. The importance of the practicum component in this course is discussed as well as the requirements for the practicum. Problems encountered in course development and future directions are also presented.

PANELISTS

Karen Billings
Logo Computer Systems, Inc.
Boston, MA

Elizabeth Carlson
University of Michigan
Ann Arbor, MI

Nancy Cunniff
Teachers College, Columbia Univ.
New York, NY

Robert Spielvogel
Learning Link - WNET/Thirteen
New York, NY

SPONSOR: SIGCUE

PROJECTS TO INCREASE COMPUTER EQUITY

Ronald E. Anderson, *Chair*
University of Minnesota
Minneapolis, MN

ABSTRACT

Computer equity is something a lot of people talk about but almost nobody does anything about. One important exception is the Equity in Technology Project administered by the Educational Computer Consortium of Ohio (ECCO) in Cleveland, Ohio. Funded by Standard Oil Company and more recently by the Ohio Department of Education, this project created a program of small grants to schools to experiment with strategies to increase computer equity for girls, minorities, and the handicapped. From this unusual program has come experience that others can utilize in planning their own programs to reduce or eliminate inequity in participation in the benefits of using computer technology. The project first will be described by members of this panel and then discussants will evaluate the implications for the educational establishment. This panel will address what needs to be done to insure that computing technology does not widen gaps among social groups.

PANELISTS

Alice Fredman
Educational Computer Consortium of Ohio
Cleveland, OH

Marlene McMillan
Cleveland City Schools
Cleveland, OH

Ellen Richman
Educational Computer Consortium of Ohio
Cleveland, OH

Jo Shuchat Sanders
Women's Action Alliance
New York, NY

SPONSOR: SIGCUE

ASSOCIATIVE BROWSING: AN INTERDISCIPLINARY APPROACH TO TEACHING HIGHER ORDER THINKING SKILLS USING MICROCOMPUTERS AND VIDEODISCS

Ted M. Kahn
Ann H. McCormick
Picodyne Corporation

ABSTRACT

This session will discuss the idea of integrating interactive computer programs, "archival" videodiscs, and curricula designed to teach creative thinking skills. Focusing on the importance of "associative browsing," we examine how previous investments in videodiscs may be maximized by the teachers' and students'use of this approach. The importance of "repurposing" different media—slides, film, audio, and videodiscs—will be discussed in relation to using hypermedia and "browsing" in developing interdisciplinary higher-order thinking skills. Several examples of existing curricula suited to this approach will be summarized.

COMPUTERS, EDUCATION, AND CHILDREN'S LEARNING: THE REAL BASICS IN EDUCATION

Thomas C. O'Brien
Teachers' Center Project
Southern Illinois University
Edwardsville, IL 62026

ABSTRACT

How shall we get there? Shall we take a train? A Bus? Pan Am? Should we walk? Maybe the best thing to do is rent a bicycle.

How shall we get there?

The obvious answer to the question is the question, "Where are we going?" If we are going to Rio, a bicycle is probably not a good idea. Nor would Greyhound do the job. If we are going around the block to have lunch, a bicycle is likely to fill the bill, as would Shanks' mares. If it's Paris, Amtrak is not a very good choice to make.

How should we get there in American education? That's the question we've been asking for so long. Should we use programmed instruction? Why not modern math? Or flash cards? Or behavioral objectives? Or performance-based contracting. No, let's use a computer. THAT will solve our problems. Or will it?

Obviously, the "How shall we get there?" sort of question is as silly an initial question in education as it is in planning a trip.

The first question is "Where are we going?" Until the question is answered, we are left with a bunch of unrelated bits and tricks and a whirlwind of bandwagons to take us there.

One major goal in education is the development of complex thinking in children.

Children's knowledge is a fabric of ideas and relationships which tends toward coherence, stability, economy, and generalizability. The goal of education should be to nutrify that fabric.

What are the actions that constitute complex thinking? Classifying, ordering, inferring, logically multiplying, reasoning propositionally, probabilistically and proportionally, keeping all but one variable constant to assess its effect, and many others.

Present day textbooks commonly pay lip service to the development of complex thinking in children but certain computer software deals with complex thinking as a major goal in children's education.

This session will be concerned with a point of view toward children's intellectual growth, and with informal demonstration, practical advice, and informal research concerning the use of such software in the classrooms.

DISCOVERING RELATIONSHIPS BETWEEN FUNCTIONS AND GRAPHS USING A GRAPHING PROGRAM

Pamala Byrd Cemen
Jerry Johnson
Department of Mathematics
Oklahoma State University
Stillwater, OK 74078
405-624-5818

ABSTRACT

Well-conceived, carefully-developed, and field-tested activities are needed in order to realize the potential of microcomputers in education. In this article,we describe assignments which guide students to discover the relationship between functions and their computer-generated graphs. These assignments, targeted at mathematics content taught in both high schools and colleges, have been developed over several semesters, assigned to thousands of students, and revised based on observed difficulties. The general format of mathematics questions and computer directions is described as well as details on each of 21 assignments. Also included are descriptions of the graphing program used, the environment in which the assignments were developed, and formative and summative evaluations. With the information in this article, high school and college teachers can write similar activities using available software and accommodating local curricular emphases.

INTRODUCTION

Computer graphics are valuable in helping students visualize mathematical concepts, such as the relationship between functions or equations and their graphs. In this paper, we describe assignments written to guide students to such discoveries. The assignments all utilize one extremely flexible graphing program, even though the topics can be found in courses ranging from beginning algebra to calculus. The content targeted is included in that part of the mathematics curriculum taught in both high school and college. Using the format described here, high school and college mathematics teachers can write similar assignments tailored to the local curriculum and computing environment.

THE COMPUTING ENVIRONMENT

The computer laboratory used for the assignments is one part of the Mathematics Learning Resource Center (MLRC) at Oklahoma State University. The MLRC is a combination computer-video-tutorial center which supplements mathematics classes at the university. The computer component contains 40 networked Texas Instruments Professional microcomputers, each with 256K, a numerical coprocessor, and a high-resolution color monitor. Each of the assignments described in this article corresponds to a specific section of the textbook being used in a particular mathematics course. In an attempt to integrate computing into the mathematics curriculum, all instructors of freshmen level (precalculus) courses are expected to give computer assignments for students to complete on their own in the MLRC. Since writing high-quality assignments is extremely time consuming, most of the instructors chose to give the ones provided. Calculus instructors were also encouraged to give assignments. Thus, each semester, approximately 4000 students are given the computer graphing assignments described here.

THE GRAPHING PROGRAM

The software used for the assignments was written by the second author. The software graphs one to ten functions in either rectangular or polar coordinates. It also permits graphing of curves defined parametrically. Functions are entered in response to a program prompt using the BASIC command DEF FNF(X); any of the functions supported by BASIC may be used. (There is also a menu of twenty-four functions that a student may choose from as well.) The package is very user friendly with copious menus, suggestions, prompts, and error traps.

The student must choose one of seven colors for each function. The choice of a color is not necessary but students enjoy being able to pick their favorite one and this makes the exercise more fun. Next the student must choose the interval for x and the number of points to be plotted. Prompts and suggestions are provided to facilitate these choices. Many graphing packages require limits on the *range* of the function; that is not so in this program: The maximum and minimum of the function are computed and used to scale both axes automatically. The resulting picture is usually satisfactory, but if not there is a menu of options to change or enhance the graph including:

- instant rescaling of the axes
- drawing a grid (rectangular *or polar*) with units chosen by the user
- moving the graph (together with the axes) closer, farther, right, left, up or down
- labeling points on the graph
- labeling points on the screen by positioning a tiny cursor.

FORMATIVE EVALUATION

During the 1986-87 school year, students spent over 1900 hours using the MLRC computers. The majority of this time was spent completing the assignments discussed in this article. This gave the first author extensive opportunity to observe the difficulties students had with the assignments and to revise them based on the information gathered.

Based on observations made during the Fall semester of 1987, these revisions have been successful in eliminating almost all of the difficulties. Although the revision process continues, our students are successful using the software with little or no assistance. In 1986-87, half of the questions asked the MLRC assistants were related to use of the software and half were related to mathematics. In Fall 1987, there are fewer questions, and a much higher proportion related to mathematics.

THE ASSIGNMENTS

After observing student reactions to the various assignments over several semesters, a useful format has emerged which is illustrated in the sample assignment shown in figure 1. The first page or two of each assignment contains the assignment itself, with no reference to the computer. The last page contains step-by-step directions for using the computer software to do that specific assignment. A two-column table guides students through the first graph. Enhancements are in paragraph form. Subsequent problems refer back to the original table, with differences explicated. Thus, students use the computer directions to get the assigned graph on the computer screen. Then, they turn back to the first page to sketch the graph on the grids provided and answer questions which lead them to discover certain properties of the graphs. Figure 2 shows a screen students would view while doing the sample assignment included in this paper.

The thoroughness and format of the directions reflect the underlying philosophy: Students should to be able to use the computers in the MLRC without any prior computing experience. Our goal is to make doing mathematics on a computer as non-threatening as possible in order to reduce mathematics computer anxiety.

As a source of ideas for teachers developing assignments, a brief description of each follows.

Beginning Algebra or High School Algebra I

Assignment	Description
Graphing Linear Equations	After completing tables of ordered pairs and graphing lines from the tables, students compare their graph to the one made by the computer. They also check their table of ordered pairs with the labeling option. The student enters the *x* value, and the program calculates the corresponding *y* value and labels the appropriate point on the graph with its coordinates.

Intermediate Algebra or High School Algebra II

Assignment	Description
Lines and Slopes	After graphing three lines with the same *y*-intercept but different slopes on the computer, students use rise/run to find the slopes from the graphs and discover that "m" gives the slope in $y = mx + b$. Next, they graph three more lines, and decide which is parallel and which is perpendicular to the first one. Finally, they find equations of lines parallel and perpendicular to a given line and check with the computer graph.
Equations of Lines	By graphing three linear equations with the same slope and different intercepts on the computer, students discover the intercept is given by "b" in $y = mx + b$. Next, they use the point-slope formula to find equations of lines through a given point which 1) have a given slope, 2) are parallel to a given line, or 3) are perpendicular to a given line. They then check their equations by graphing them on the computer and using the labeling option to make sure each line goes through each given point.
Systems of Linear Equations	Students use the symbolic manipulation package muMATH[1] to solve a system of linear equations algebraically and the graphing program to solve the same system graphically. With the labeling option, they can see that the coordinates of the point of intersection are the same as the values of *x* and *y* in the algebraic solution. Also, they examine computer-generated graphs of linear systems of equations to decide whether each system is dependent, independent, or inconsistent.
Quadratic Functions	Students graph quadratic functions and discover that, if the coefficient of x^2 is positive, the resultant parabola opens upward, and if it is negative, the parabola opens downward. Next, they find the vertex and the *x* and *y*-intercepts of computer-graphed quadratic functions.

College Algebra or Precalculus

Assignment	Description
Properties of Graphs	For each of four functions graphed on the computer, students decide 1) whether it is even, odd, or neither, 2) the range for an infinite domain, and 3) the intervals of *x* where the function is increasing and where it is decreasing.

Shifting, Stretching and Reflecting Graphs	Students use the computer to graph sets of three functions. In each set, the second and third can be obtained from the first by either shifting, stretching, reflecting or a combination of these. Students must tell which has occured and, when shifted, how many units and in which direction.
Inverse Functions	Students decide whether two functions are inverses by seeing if their computer-generated graphs reflect each other in the line $y = x$. Also, they find the inverse of functions algebraically and use the computer graphs to check.
Graphing Quadratic Functions	Students find the vertex, axis of symmetry, maximum or minimum function value, range, and x and y-intercepts of quadratic functions they have graphed.
Polynomial Functions	By viewing computer graphs of polynomial functions of the form $f(x) = ax^n$ for $n = 2$ to 5, students discover these functions are symmetric about the y-axis when n is even and symmetric about the origin when n is odd. In the second part, they find the x-intercepts or zeros, the intervals where $f(x) > 0$, and the turning points (inflection points) from computer-generated graphs of polynomial functions.
Rational Functions	Students find horizontal and vertical asymptotes and x and y-intercepts of rational functions from their graphs.
Graphing Exponential and Log Functions	After graphing exponential and logarithmic functions on a computer, students 1) find horizontal or vertical asymptotes, 2) decide whether the given function is increasing or decreasing, 3) find the x or y-intercept, and 4) find the domain or range. They also explore shifting and inverses of these functions.
Systems of Linear Equations	Students use muMATH to solve a system of linear equations with matrices and the graphing program to solve the same system graphically. With the labeling option, they can see that the coordinates of the point of intersection of the lines is the same as the x and y values in the matrix solution.
Ellipses and Hyperbolas	Students find the coordinates of the center and the endpoints of the major and minor axes from the computer-generated graphs of two ellipses. Next, they find the equations of the aysmpototes of each of two hyperbolas and check them by graphing.

Trigonometry

Assignment	Description
Graphing Trigonometric Functions	After graphing sets of three sine or cosine functions on the computer, students discover which constant in $A\cos(Bx + C) + D$ varies the period, which varies the amplitude, which does phase shifts, and which shifts the function up or down.
Graphs of Other Trigonometric Functions	Students explore the relationship between 1) secant and cosine, 2) cosecant and sine, and 3) tangent and cotangent. They discover, for example, that the zeroes in one produces asymptotes in the other.
Addition of Ordinates	By examining computer graphs of two functions with their sum, students discover that the coordinates of the sum function is the sum of the coordinates of the individual functions at each point. At least one function of the original pair is trigonometric.
Polar Equations	Students explore graphs in polar coordinates to discover what variations produce graphs of roses, cardioids, or "near circles". This is also a useful way of better understanding trigonometric functions.

Calculus

Assignment	Description
Continuity	After graphing a piecewise function with jumps and holes in the graph, on the computer, students find limits and one-sided limits, and decide where the function is or is not continuous.
Limits and Asymptotes	Students find infinite limits, limits at infinity, and asymptotes from computer-generated graphs of rational functions.

Curve Sketching	Students use muMATH to find the first and second derivatives of a function and to evaluate them at specific points in order to 1) find local extrema, 2) determine where the function is increasing or decreasing, 3) find inflection points, and 4) determine where the function is concave up and concave down. After using this information to sketch the graph, they can check their work with the graphing program.
Exponential and Log Functions	Students find infinite limits and limits at infinity from computer-generated graphs of exponential and logarithmic functions. They also find inverses of logarithmic functions and use the graph to check.

SUGGESTIONS FOR DEVELOPING ASSIGNMENTS

Following are several suggestions you may want to keep in mind when writing assignments of this type.

1. Find functions and intervals which show up well with your software. For example, since our program scales the *x* and *y*-axes the same, if the range is much larger than the domain, the graph appears extremely small on the screen. Also, certain intervals make the grid too large or too small. Finally, some programs "jump" over vertical asymptotes given the right interval and number of points.

2. Unless your students are experienced at using computers, provide thorough, step-by-step directions using the two-column format shown in this article.

3. Provide grids for students to use to sketch the graphs from the screen, making sure the grids provided match the ones generated by the computer. In addition to saving money on printing supplies, sketching the graphs from the screen provides a valuable learning experience.

4. Be creative in figuring out how to enter functions in BASIC. For example, solve conic sections such as $x^2 + y^2 = 4$ for *y* and enter in two parts, the positive and negative square root. Also, use the base changing formula to enter logarithms in bases other than *e*.

SUMMATIVE EVALUATION

During the Fall semester of 1987, the MLRC assignments are being evaluated in terms of their effectiveness in helping students learn the target concepts. College algebra (precalculus) students who actually do their assignments in the MLRC are being compared to those who do not do them in the MLRC with a pretest-posttest design. Unfortunately, since this is a real setting, rigorous experimental control is impossible. Instead of random assignment of students to the two groups, students self-selected whether or not to use the MLRC. All students in the study are in classes where the assignments will be collected, graded, and counted as a quiz grade. Results will be provided when they are available.

CONCLUSION

Computers and mathematics laboratories have tremendous potential for influencing the teaching of mathematics. The extent and value of this influence will depend on how creative we are in exploiting these facilities. One way to do this is by designing innovative exercises that tap this exciting resource. The examples presented here are an effort to address this need.

NOTES

1. muMATH is a symbolic manipuation software package which performs mathematical operations such as solving equations, taking derivatives or integrals, and row reducing matrices. It can be purchased from Soft Warehouse, 3615 Harding Ave., Suite 505, Honolulu, Hawaii, 96816.

Figure 1. Sample Assignment

Math 1613 Section 3.3: GRAPHING TRIGONOMETRIC FUNCTIONS **Fall 1987**

Instructor______________________ Name______________________

Due Date______________________ Points______________________

For each of the problems below, graph the three functions on a single set of axes, using the interval $[-3\pi, 3\pi]$. Then answer the questions.

1. $F(x) = \cos x$ $F(x) = \cos 2x$ $F(x) = \cos 4x$

   The period of a function is the interval of x covered before it repeats.

   What is the period of $\cos x$? ______________________

   What is the period of $\cos 2x$? ______________________

   What is the period of $\cos 4x$? ______________________

   What do you think the period of $\cos Bx$ would be? ______________________

2. $F(x) = \sin x$ $F(x) = 2\sin x$ $F(x) = 3\sin x$

   The amplitude of a function is the maximum distance it reaches from the x-axis; that is, the distance from the highest or lowest point to the x-axis.

   What is the amplitude of $\sin x$? ______________________

   What is the amplitude of $2\sin x$? ______________________

   What is the amplitude of $3\sin x$? ______________________

   What do you think the amplitude of $A\sin x$ would be? ______________________

3. $F(x) = \cos x$ $F(x) = \cos x - \pi$ $F(x) = \cos(x + \frac{\pi}{2})$

   Where is $\cos x - \pi$ in relation to $\cos x$? ______ units ____________

   Where is $\cos(x + \frac{\pi}{2})$ in relation to $\cos x$? phase shifted ______ units

   Where would the $\cos(x + C) + D$ be in relation to $\cos x$ if $D < 0$?

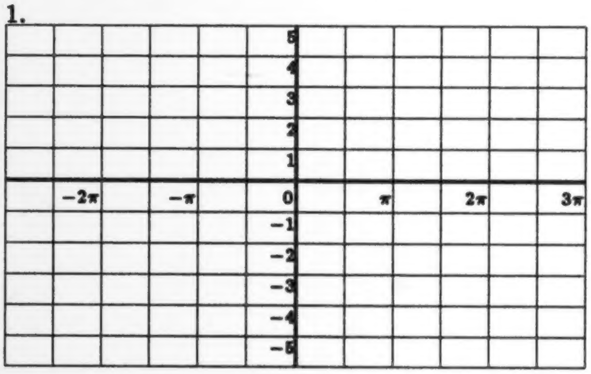
   ______ units ____________ and phase shifted ______ units

4. $F(x) = \sin x$ $F(x) = \frac{1}{2}\sin(x - \pi)$ $F(x) = -2\sin 2x$

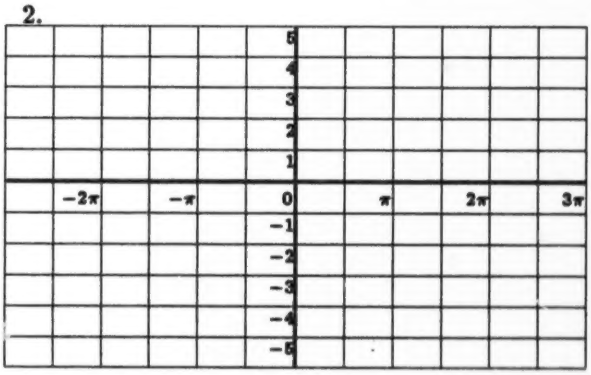
   What is different in the second graph relative to the first one? ______________________

   What is different in the third graph relative to the first one? ______________________

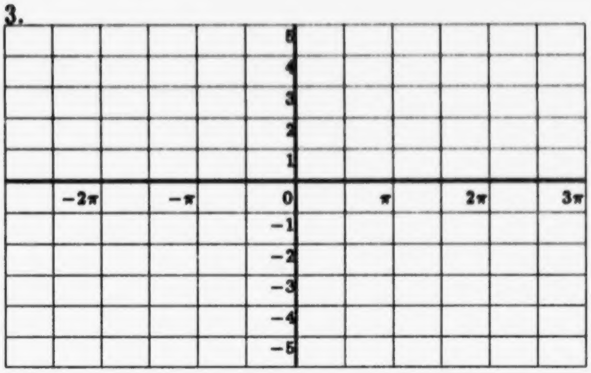
1.



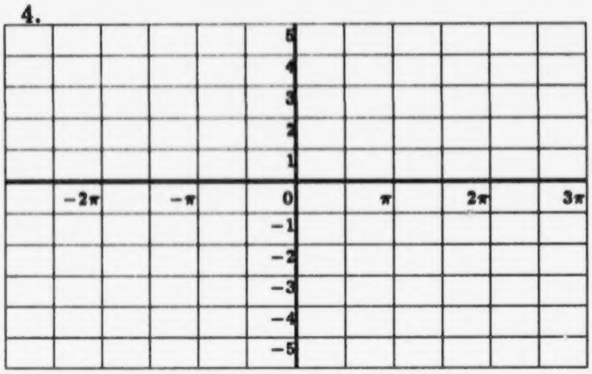
2.



3.



4.



Using the MLRC Computers to Graph Trigonometric Functions

1. Check out the Network Disk (TI) in the MLRC. Choose a computer. Put the disk in the left drive, close the lever, and turn the computer on (switch is on right in back). On the first menu, press 1 and [RETURN] to log on to the network. On the MLRC menu, type 1 for graphing.

2. On the Graphing Menu, select option B for "Graphing Several Functions". Do not press the [RETURN] key. Respond as follows.

Computer	Your Response	
"What do you want to graph?"	1	
"Number of functions?"	3	[RETURN]
"Enter function # 1"	COS(X)	[RETURN]
"Choose color"	Enter a number	
"Do you wish multiples of pi?"	Y	
"Input interval"	−3,3	[RETURN]
"Number of points?"	200	[RETURN]
"Enter function # 2"	COS(2*X)	[RETURN]
"Choose color"	Choose a different color	
"Enter function # 3"	COS(4*X)	[RETURN]
"Choose color"	Choose a third color	
BEEP	[RETURN]	

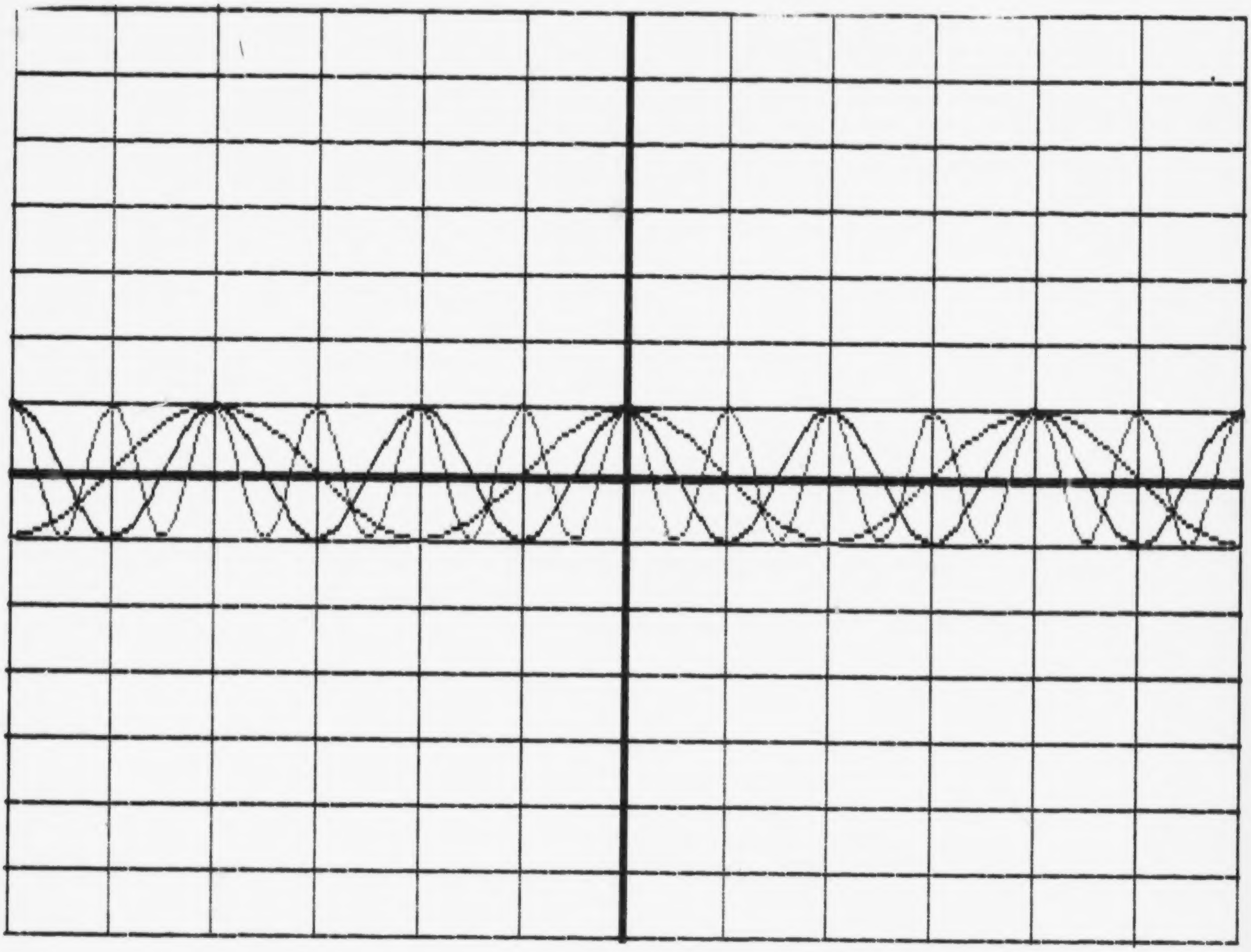
3. Draw grid. For the menu of options, press the [SPACEBAR]. Type **G** for grid, **R** for rectangular coordinates and **1.57,1** for the units. Use the grid to sketch the graph and answer the questions.

4. When you finish the first question, type **F** for graph more functions and **Y** to confirm. Follow the procedure used above, except for entering the functins as indicated below.

Problem #	Function #1	Function #2	Function #3
2	SIN(X)	2*SIN(X)	3*SIN(X)
3	COS(X)	COS(X) − 3.14	COS(X+1.57)
4	SIN(X)	1/2*SIN(X−3.14)	−2*SIN(2*X)

5. Log off. When you are finished, press the [SPACEBAR] for the options menu. Type **Q** for quit, **M** to exit to the master menu, **F** to exit to the main menu, and **7** to exit to DOS. When you get back to the original menu, take the disk out and turn the computer off.

Figure 2. Screen display of *cos x*, *cos 2x*, and *cos 4x* on $[-3\pi, 3\pi]$ with a $\frac{\pi}{2}$ by 1 grid.



TEACHING BASIC AND PASCAL "LOGO STYLE"

Sharon Burrowes Yoder
Logo Computer Systems, Inc.
330 West 58th Street, Suite 5M
New York, New York 10019

ABSTRACT

In traditional programming courses assignments usually consist of problems that are somewhat closed in nature. That is, students are usually given a rigidly defined problem statement and are provided with a set of data that they are to use to test their program which will produce a unique result. While this approach makes evaluation easy, it discourages creativity and exploration, and encourages inappropriate sharing of information.

Another approach to assigning programming problems is to use open-ended assignments that encourage students to make the problem their own and include their own creative ideas in their work. Such projects require that the teacher examine carefully the objectives for a particular assignment and plan the evaluation scheme thoughtfully.

Student response to such open ended assignments is quite positive and the resultant work is extremely varied and often very creative. An important side effect of such assignments is that group interactions are much more appropriate. Students help debug each other's projects rather than simply trading algorithms. Further, students learn from each person with whom they interact, thus extending their knowledge well beyond that acquired when solving a single solution problem.

INTRODUCTION

The core of most computer programming instruction, whether it be embedded in the elementary curriculum, part of a junior high module, or semester or year long high school or university courses, is the hands-on assignments that require students to make use of the programming language that they are learning in the classroom. Assignments for such courses tend to be carefully specified, usually requiring that the finished product behave in a rigid manner, often using a specific set of data.

This approach, while it does require students to make use of the computer to solve problems, has several weaknesses. First, it allows for little or no creativity or exploratory learning. While the students may choose somewhat different algorithms for solving such problems, there are usually a limited number of methods of solution, especially with problems appropriate to beginning programmers. Second, all too often only a few of the best students actually solve the problem assigned; the remaining students merely adapt the algorithm developed by the few. Sometimes this is conscious cheating on the part of students; at other times, those seeking help simply end up getting too much assistance. In either case, the often implicit goal of teaching problem solving is quickly lost for the majority of students.

The Logo philosophy offers an alternative way to think about teaching programming. Ideally, students in a "Logo environment" are encouraged to develop their own projects and to use exploratory learning and discovery to reach their goal. It is possible to merge this open ended, discovery approach to the learning of programming with the traditional approach to teaching any programming language. This can be done in such a way that students become more engaged in what they are doing while maintaining standards of learning that are typically required in a programming course.

The techniques for the teaching of programming to be described in this paper were developed over a several year period in a semester Introduction to Computers and Computing course offered to students in grades 9 - 12 in a small-town high school in the Midwest. These ideas were more recently applied to other high school courses including courses in Basic and Pascal, Advanced Placement Computer Science, and to teacher training courses. While the fundamental technique has remained unchanged, problem sets and evaluation schemes continue to evolve as they are used with more students and in a variety of courses.

The ideas for giving open-ended assignments grew out of the Logo philosophy, put forth most clearly by Seymour Papert in *Mindstorms: Children, Computers, and Powerful Ideas*. The "Logo environment" encourages open-ended projects and exploratory learning. Students involved in such self-directed projects become more deeply engaged in their work because their goals are their own, rather than being imposed by an outsider.

A few years ago, the character of the students enrolled in the introductory high school computer course taught by this author changed from primarily "hackers" to students interested in a more general introduction to computers. Traditional rigid problems designed to prepare students for college-level computer science courses seemed less and less appropriate. Initially the techniques described in this paper were used with an occasional assignment. As the techniques became increasingly successful with students in a limited context, they were tried with a wider range of assignments and then with increasingly advanced students as well as inservice teachers, all with equal success.

BACKGROUND

Anyone who has taken a traditional programming course has probably encountered a problem such as this:

> Write a program to compute the take-home pay for an employee of the Acme Computer company, given the hourly rate of pay and the number of hours worked. Any hours over 40 should receive time and a half. Run your program three times, once for each of the following sets of data: $5.25 for 23 hours; $9.95 for 51 hours; $7.53 for 40 hours.

Once a student has the necessary assignment statement in hand, whether it is acquired through individual problem solving techniques or from a friend, "solving" the problem becomes trivial.

Beyond the process of creating a program that runs, the instructor assigning such a problem no doubt has clear objectives in mind. Likely, in the case of the problem above, the objective is to have students use the assignment statement, conditionals, and arithmetic operators.

A Pascal textbook that has been used at both the high school and university level provides another typical problem:

> The input consists of groups of three numbers. The last group is an end-of-list signal consisting of three zero values. Write a program to read each group in, print it out, and print an indication of whether the three numbers represent the sides of a triangle. (Conway, Gries, and Zimmerman, p. 85)

What is the objective implied by this problem? Since it follows a chapter on loop structures, most certainly the authors expect the student to use either the Repeat-Until or While-Do construct. Further, it is clearly asking students to deal with an end of data condition that is not trivial.

From a more recent Pascal textbook comes a problem that acknowledges the sameness of so many programming problems:

> In consideration of all the business programs that compute retail prices, write a consumer's program that figures out wholesale prices. Given a purchase price, it should deduct a 40% markup, and a 5.5% sales tax. Be sure to make the deductions in the right order. (Cooper and Clancey, p. 67)

It appears that the authors of this text were uneasy with the traditional programming assignments, but did not completely break away from the mold, even though this book contains many problems that are much more interesting and challenging than those found in most programming books.

Clearly, the programming problems described here require the student to use specific aspects of the language being learned. The instructor can easily evaluate whether the program is correct or not by having the student run the program on a specific data set. Matters of program design and style can be dealt with by examining the program listing. Evaluation is relatively unambiguous and there is little that is subjective in the grading process.

METHODOLOGY

The first step in moving away from traditional programming assignments is the creation of open-ended problems that encourage discovery learning. This is perhaps the most difficult part of the teacher's task. It is important to be very clear about the objectives to be met before designing each assignment. An example of such an assignment is:

> Write a program that uses music and graphics or music and text. Your program should have music playing while some action takes place on the screen. Be careful when designing your program so that the subdivision of your procedures make sense. Extra credit will be given for using music, graphics, and text.

Beyond making use of commands to create music and graphics, the fundamental objective for this particular assignment is to have students use good modular design in their programs. In order to have music accompanying another action and avoid excessively long procedures, the students must think carefully about how to subdivide the program. The actual student programs that result from this assignment are quite varied. One student will produce an animated picture with musical accompaniment. Another will create a sing-along by presenting text as music is played. Yet another will create an adventure story with appropriate accompanying music. No two programs are ever exactly alike.

The second problem facing the teacher is how to evaluate such varied results. Before presenting an open-ended assignment, the teacher must think carefully about the objectives to be met and how each will be evaluated. In addition, there must be clear guidelines as to what proportion of the grade will include such factors as programming style and form of output as well as what proportion will include meeting each of the objectives of the assignment. Often a check list such as the one shown in Figure 1 is helpful is such an evaluation.

RESULTS

The most obvious result of this change in type of assignments is that students become more engaged in their projects. No longer is the computer programming assignment just an exercise to complete; rather it becomes more personalized. Students take great pride in their accomplishments when they can inject their own personal touches into their work.

What is initially less obvious is the ways in which working with others changes. In a traditional environment, the most able students are quite "popular" just before an assignment is due. Students who rely on their more capable peers in this way, simply do not learn much by solving any particular problem. When students generate their own projects, there is no longer a tendency to ask others for "the answer". Instead, the quality of interactions changes. Students share techniques, ideas, and help each other debug projects, in the process often learning more than just completing only their own work.

Evaluation of these open-ended projects is certainly more interesting for the teacher. One never tires of grading the same code over and over, and the creativity of students generates constant surprises. However, the process of evaluation is quite different. While considerations such as programming style and the use of modularity in a program likely remain much the same, simply checking the results of a data set is no longer possible. Each project must be examined carefully to determine whether the objectives were met. In the case of the sample problem given above, the teacher would have to run each program to see if, indeed, the music played while text or while text or graphics appeared dynamically on the screen. This can be a time-consuming process.

Evaluation of open-ended projects often results in more success by students. Seldom does a student turn in an assignment that is "wrong". If the project does not meet the objectives set forth by the assignment, it is usually the result of a misunderstanding on the part of the student. This is often a place to apply mastery learning techniques, and allow the student to attempt the project again. Since each student's work is unique, there is no concern that the student will simply copy another's work.

The time involved in evaluating such work is well worth the effort. Knowing that students are both enjoying their work more and developing better individual skills is powerful motivation for the concerned teacher. As the instructor's skills in using this new method improve, the time involved decreases markedly. One learns how to quickly find salient features in a program and to specify problems in such a way as to minimize difficulties at the point of evaluation.

SUMMARY

At the heart of this discussion is the question "Why do we teach programming?" Traditionally, the goals of an introductory programming course include teaching the syntax and semantics of the language, teaching control and data structures, and providing some insight into the discipline of computer science. Giving open-ended assignments in no way detracts from these goals. In fact, even more objectives can be met with the techniques described in this paper. This approach provides for individualization, increases motivation of students, encourages creativity, discourages inappropriate assistance, encourages positive group interactions, and provides for more success experiences for students.

Further, these techniques are quite effective with students at all levels. Simply telling more advanced students that they are to implement a stack or linked list in a meaningful way requires students to practice a new technique, but allows them room for individual expression.

In a time when our overall goal is to have students be comfortable with the computer, to make the technology a part of their lives, it seems far more important to encourage students to use the computer creatively rather to insist that students march lock-step through traditional and often meaningless exercises. The result of such changes would likely be a generation of students more comfortable with the computer as an integral and useful part of their world.

REFERENCES

Amsbury, Wayne, *Structured Basic and Beyond,* Computer Science Press, Potomac Maryland, 1980.

Conway, Richard, Gries, David, and Zimmerman, E. C., *A Primer on Pascal,* Winthrop, Cambridge, Massachusetts, 1978

Cooper, Doug and Clancey, Michael, *Oh! Pascal,* W.W.Norton, New York, 1982.

Papert, Seymour, *Mindstorms: Children, Computers, and Powerful Ideas,* Basic Books, New York, 1980.

-GRADING SHEET FOR LOGOWRITER PROJECTS USING PROCEDURES

NAME_________________________ PROJECT NUMBER ______ CLASS PERIOD _______

****

INSTRUCTIONS:

ALWAYS:
- put a procedure tree on the back of this page
- use PRINTTEXT80 to print the flip side of your LogoWriter page
- staple the listing of procedures to this sheet

IF YOU ARE TO TURN IN A RUN:
- run the program to demonstrate that it works
- staple a print of the front of your LogoWriter page to this sheet

IF YOU ARE TO PUT YOUR PROGRAM ON DISK, fill in each blank below:

DISK NAME: ________________________
PAGE NAME: ________________________
PROGRAM NAME: ______________________

****

STYLE (35%)

Meaningful procedure/input/variable names---------------------------(5) _____%
Procedures not chained; program uses only user defined proc. calls--(5) _____%
One statement per line; no unnecessary rewriting of code------------(5) _____%
Procedures/lines of reasonable length: not too short; not too long-(5) _____%
Procedures well formatted---(5) _____%
Uses good overall programming style--------------------------------(10) _____%

PROBLEM SOLUTION (25%)

Procedure tree included on the back; procedure tree correct---------(5) _____%
Flip side contains only procedures used in the program--------------(5) _____%
Logic used in solution is straightforward/can be generalized--------(5) _____%
Modularity; procedure names reflect procedure functions------------(10) _____%

COMPLETE AND CORRECT RESULTS (25%)---------------------------------(25) _____%
(Program produces the results required by the project statement)

OUTPUT (15%)

Good use of page; correctly formatted; easy to read-----------------(5) _____%
User friendly; easy to understand purpose of program----------------(5) _____%
Front of page blank or contains appropriate information-------------(5) _____%

EXTRA CREDIT (20%)

Embellishments (above and beyond minimum)---------------------------(5) _____%
Approved extra enhancement---(15) _____%

****

COMMENTS:

Total percent = _____%
times maximum possible points = * _____ = _____points
less days late * 5 = _____ * 5 = - _____points
Final score (out of _____) = _____points

___See the flip side of your page (on your disk) for comments.

___See your attached pages for comments.

___See above for comments

Figure 1

TELECOMMUNICATIONS

Margaret Bingham
M. Holloway
C. Denning
J. Campbell

ABSTRACT: Computer Assisted Retrieval (C.A.R.) Project: Instructional Use of Dow Jones

Margaret Bingham

Four school systems were provided a subscription to Dow Jones News Service, portable computers and printers, sample instructional projects, and training. After using the online service for several months, they determined that the activity was appropriate for certain areas of the curriculum. The project goals, cost figures, staff training components, evaluation results, and the lesson plans developed will be discussed.

ABSTRACT: Dialog and Einstein: Extending the Media Center Resources

M. Holloway

The DIALOG Classmate Program was tested at a senior high school (grades 9-12) with 2200 students and a media collection of 27,000 volumes and 108 magazine subscriptions. Project results including costs, equipment, staff and student user skills, and resource benefits will be shared.

Two school systems used the information gateway service, Einstein, formerly from Addison-Wesley. Both high school sites were located in rural areas with no previous online experience but a need to explore providing resource access for their students beyond the walls of the media center. The staff training, the search projects, and the benefits of a gateway service will be outlined.

ABSTRACT: Project DIAL: Developing Instructural Activities Online

C. Denning

Project DIAL was implemented to discover the contributions that could be made to the instructional program through the use of telecommunications. Three different thrusts were explored: telecommunications within a school system, telecommunications within the state, and telecommunications between states. The benefits and implementation needs of telecommunications in the areas of writing and social studies resulted as the first group published a system-wide newspaper, the second group (eight schools) worked in pairs to produce project reports, and the schools in the third group exchanged cultural information with Maine, New York, Montana, Massachusetts, and Vermont.

ABSTRACT: Statewide E-Mail: Communicating Instructional Activities

J. Campbell

The implementation of a statewide electronic mail system, using FrEdMailer, will be presented, including cost figures, training guidelines, and use benefits. Sample instructional activities which have been successful will be described.

TEACHER EDUCATION

Doris Carey
Carolyn Jacobi
Mildred Lintner
Anne Colman, Helen Hartle, Stan Silverman

ABSTRACT: Graduate and Undergraduate Curriculum in Methods of Teaching Computer Science

Doris M. Carey, College of Education, University of Nevada, Las Vegas, NV

Curriculum for classes in the methods of teaching computer science must take into account the background and needs of the learner, the requirements of certification and accrediting agencies, and the standards and guidelines of educational institutions offering the courses. Curriculum for undergraduate courses in the methods of teaching computer science can be modeled after methods courses in other subject areas. Course goals would typically include long range planning of curriculum units, the refining of skills in writing lesson objectives, the planning of individual lessons, the design of lessons based on the use of a variety of teaching strategies, classroom management strategies that take into account the specific requirements of the subject area, the evaluation of teaching and learning, approaches to teaching problem solving, and the effective use of equipment and materials.

Graduate courses in the methods of teaching computer science are expected to accomodate the experienced teacher. The curriculum for such courses should include advanced models of teaching such as concept attainment, inquiry, concept formation, and inductive models. Curriculum planning and evaluation, performance-based testing, and problem solving strategies are topics that lend themselves to a graduate treatment of computer science teaching. A seminar format for at least part of the course would provide a forum for the sharing of effective strategies and teaching ideas such as student projects and student-centered teaching models. Participants might also be involved in discussions comparing the teaching of programming with the teaching of computer science.

Current trends in the restructuring of teacher training programs indicate that graduate-level courses will be needed for inexperienced teaching candidates preparing for an internship or induction year. Clearly, courses offered at this level would differ from the undergraduate curriculum in that a higher degree of rigor and a more advanced treatment of the topics would be required. Such a course might include the investigation and application of research on teaching to the teaching of computer science. Furthermore, a graduate-level teacher preparation program might assume that students hold an undergraduate degree in computer science--far more coursework than is currently expected of teachers of computer science in most states. An emphasis on sequencing of topics, writing of objectives, teaching strategies, and evaluation would be required, within a structure and sequence that takes into account the development of skills within the field component.

ABSTRACT: K-6 Computer Programs; A Sensible Solution

Carolyn Jacobi, Educational Technology Specialist, Rapid City Area Schools, Rapid City, SD

Our K-6 computer program - "A Sensible Solution" - is just what the name indicates. Drawing from many sources, adopting those things that worked from already exising K-6 programs, and filling in the spaces from the research, we initiated a workable, sensible, successful program which combines in equal parts the training of teachers and students, and integrating technology into the existing curriculum.

Employing professionals who were experienced elementary teachers, who had gained a lot of credibility with their peer group, and who were also curriculum developers that were state accredited in computer, the curriculum was scanned for obvious, but also imaginative ways in which to infuse and use technology. At the same time, grade-appropriate skills for students were identified and a teacher training plan was also developed.

It was determined that the final product be put on a three year implementation with the computer assisted activities gradually shifting from a "computer teacher" responsibility to the classroom teacher as their skills gradually developed.

During the first year of implementation, the knowledge-based applications software is used in a lab setting with an entire classroom. The "computer teacher" provides demonstration teaching and all materials, leaving the classroom teacher free to discover the software with their students, take notes on the presentation, or scan additional material.

In the second year of operation, the responsibility of what is being taught in the computer lab is shared by the computer teacher and classroom teacher.

By the third and last year of implementation it is anticipated that classroom teachers will be capable of making the proper judgements for the use of technology in their curriculum and classrooms. In the meantime, the students have had the benefits of an integrated curriculum, properly taught and supported by teachers who understand both instructional computing and the district curriculum.

The bottom line, however, is that this program is enthusiastically accepted because it is centered on the nucleus of the class structure and within the appropriateness of that class's instruction.

ABSTRACT: What's Happening in Computer Science Teacher Certification

Dr. Mildred D. Lintner, Department of Computer Science, Eastern Michigan University, Ypsilanti, MI 48197

In the past few years, computer science has become more and more recognized as a distinct discipline in the secondary school curriculum. As a result, there has been growing concern over the quality of both computer teacher preparation and computer curriculum content. This concern has manifested itself in two ways: (a) A growing number of state boards of education have proposed and are implementing computer teacher certification requirements; and (b) statewide curriculum guidelines are being developed to prescribe the content of computer education courses.

In June 1987, at a SIGTE meeting during NECC '87, a committee was formed to find out what was going on in computer science teacher certification across the nation. The committee, all faculty members at teacher education institutions, proposed to survey a broad sampling of state boards of education, asking the following questions:

1. Is there Computer Science Teacher Certification or Certification Endorsement in this state?
    a. If so, when was it instituted?
    b. If not, is such certification being planned or contemplated?

If certification either exists or is on the planning board:

2. What constitutes certification requirements for new teachers?
    a. What constitutes certification for existing teachers?
    b. What provisions, if any, are there for "grandparenting" teachers with previous computer science teaching experience?
3. What constitutes "computer science" in your state?

This discussion will present the findings of the committee's survey. We expect to be able to present and compare certification requirements of numerous states and derive from our comparisons an indication of nationwide trends in this area. We expect to be able to report on the experiences of those states which have already adopted their computer certification requirements and have been implementing them for a period of several years. We expect to be able to report on what issues and debates have been delaying certification implementation in states where a program has not yet been adopted. And finally, we expect to be able to derive some guidelines for teachers and prospective teachers having to face certification over the next few years.

ABSTRACT: Computer Conferencing a Teacher Assistance Model

Anne Colman, Director, Sullivan County Teacher Resource Center
Dr. Helen Hartle, New York State Education Department
Stan Silverman, Director, Educational Technology, New York Institute of Technology

The Teacher Resource and Computer Training Centers in NYS were established by NYS to provide systematic and ongoing professional support services to teachers (91 centers addressing over 179,000 teachers). In order to make appropriate use of experts the Teacher Centers in cooperation with the New York Institute of Technology has developed and implemented an extensive telecommunications linkage which provides both center to center communications as well as direct teacher support.

The Centers are connected to a VAX cluster at NYIT hosting TECHNET (Computer Conferencing, Electronic Mail, VideoText and Terminal Phone) through 800 phone line service. The centers through a telecommunications advisory committee have established an extensive telecommunications infrastructure. This structure includes over 20 separate conferences covering all aspects of center work and all the content areas.

Fundamental to what it is that TRCs do is the atmosphere of mutual respect, conducive to self-direction and collaboration, which they provide. TRCs provide safe harbors which permit a level of validation and experimentation essential for good growth. By overlaying technology on this already effective strategy, a mechanism is provided that enables the dialog to continue asynchronously and without regard to the constraints imposed by distance and geography.

A major goal of this project is to provide support and instruction to teachers using the vehicle of telecommunications. Although the pilot project was specific to the concerns of teachers of *at risk youth*, the model has already been expanded to include diverse populations of teachers.

Specifically the model consists of a computer teleconferencing environment which contains the following elements:

1. A Faculty Lounge - a place where teachers can EYE LEVEL with other teachers who share a common agenda.
2. Expert Conference - a place where teachers can ask questions, discuss and develop ideas with other teachers who have had success in a given area.
3. Research Conference - a place where teachers can discuss issues, pedagogical models and research with current University based researchers.
4. Clearinghouse Conference - an online full text searchable database of human and material resources.

SIMPLE AND INEXPENSIVE COMPUTER BASED INSTRUMENTATION

Bill Rouse
Professor of Computer Science Education
Miami University
Oxford, OH

ABSTRACT

Computer based instrumentation refers to the practice of using computers to detect, measure, and monitor physical phenomena. Although computer based instrumentation systems specially designed for school applications can now be purchased in either ready to use or kit form, it is very simple and inexpensive to implement computer based instrumentation in the form of teacher-made sensors, attached to the game port of a microcomputer. Such sensors can be adapted to any microcomputer that possesses a game port, but this paper describes adaptations for the Apple II.

The game port connector is a socket located at the right rear corner of the main circuit board inside the Apple II+ or IIe. In addition to other input and output connections this socket contains four separate game controller input connections, numbered 0, 1, 2, and 3. In order to use these game controller inputs for instrumentation purposes, it is not necessary to understand how they work, but only to understand what they do. The game controller inputs were designed to interact with game controllers or "paddles" so that users could play "pong" games. A paddle is nothing more than a 150 K ohm rotary linear-taper potentiometer (a variable resistor). As the knob on the potentiometer is turned, the computer reacts to the varying resistance by generating a whole number in the range of 0 to 255. The number generated is proportional to the value of the resistance of the potentiometer. A BASIC programmer can determine the number generated as a result of the current setting of a game controller by using the PDL command. Thus LET NUM = PDL(0) assigns the number from game controller 0 to the variable NUM, or PRINT PDL(1) displays the number from game controller 1 to the screen.

Since the computer can't tell what kind of device is attached to a game controller input, any device can be attached and used provided it generates a resistance that varies within the range of 0 to 150 K ohms. Thus, any device that generates a resistance in this range and that varies in proportion to the phenomenon to be measured, can be used as an instrumentation sensor.

This presentation will provide directions for the construction and use of a displacement detector, a temperature probe, and a photogate, and will include AppleSoft BASIC listings to enable these sensors to sense the displacement of objects, measure temperature, count repetitive actions, time events, and measure velocity and acceleration. It will be supplemented with a handout that can be duplicated and shared with science and computer science teachers in your school system.

Certain instrumentation applications require more than four sensors. Others require types of sensors that cannot easily be adapted to the game port. For such applications more complex and expensive systems must be interfaced directly with the computer bus. However, as this presentation will demonstrate, the game port approach has the advantages of simplicity and low cost and yet effectively satisfies many instrumentation needs.

A YEAR-LONG NATIONWIDE FIELD EXPERIMENT ON THE EFFECTIVENESS OF COMPUTER USE IN GRADE 5 TO 8 MATHEMATICS CLASSES

Henry Jay Becker
Center for Research on Elementary and Middle Schools
Johns Hopkins University

ABSTRACT

Informal observations of students using computers suggest that students are enjoying their experiences and that they are making progress in their understanding and skills. But skeptics reply, first, that enthusiasm does not inevitably lead to increased learning; second, that some students would have made progress anyway; and third, that computer-using teachers would be successful even without having access to computers.

To resolve unanswered objections, and indeed to learn what circumstances do lead to improved achievement (and for which types of students), we need systematic evaluations in typical school settings that contrast achievements by students given substantial amounts of computer-based learning time with achievements of comparable students taught using only "traditional" (non-computer) media and taught by the same, or a similarly qualified, teacher.

This session presents initial findings from the first of several planned field experiments covering computer use in different school subjects in the fifth through eighth grades. This first experiment measures changes in mathematics achievement and attitudes over a full year of instruction in 48 pairs of classes in 31 different schools from 16 states around the country. The field period lasted from October 1987 to May 1988.

In each pair of classes, one class of students was randomly assigned to be taught math with extensive use of computers; the other class used no computers for mathematics. The students were either randomly or systematically assigned to classes so that each class in a given pair of classes had a similar mixture of students. Most teachers taught both a "computer" and a "traditional" class. In schools with self-contained instruction, two computer-experienced teachers were randomly assigned to one of the "treatments."

In all but one school, computer-using classes in the study had at least one computer available for every four students in the class. Most students used computers for an hour or more each week throughout the year. The curriculum followed and the software used to support it was determined by each school independently. Most teachers emphasized computation in their math instruction, and this was reflected in their choice of software as well. However, a substantial fraction of the classes emphasized word problems, logic, and other problem-solving activities. The most commonly used software was Milliken Math, Milliken WordMath, the IBM Math Practice/Math Concepts series, Sunburst's Kings Rule and other problem-solving software, MECC's Number Munchers and other MECC disks, the University of Northern Iowa's Microcomputer Curriculum Project, S.R.A. Math, Houghton-Mifflin's Math Activities Courseware, and Scholastic's Math Shop.

Data was gathered throughout the year on the curriculum that each teacher followed, the computer programs used, and organizational aspects of how they were used. Questionnaire and achievement data were collected in the Fall and again in the Spring. Achievement tests used include a standardized test and a curriculum-specific test developed for each pair of classes to measure achievement over the domain of instruction in that pair ("computer" plus "traditional").

The data presented during this conference includes a description of the students' and teachers' backgrounds and the curriculum followed by each teacher as well as an analysis of some of the achievement data. Several of the teachers participating in the study describe their own implementations and discuss factors that might have affected the achievement scores in their particular setting. The presentation also indicates future analyses to be done--for example, analyses of the relative effectiveness of computer-based approaches that emphasize computation versus those that emphasize problem-solving or the benefits for generally low-achieving students versus the benefits for higher-achieving students.

PANELISTS

Teachers participating in the study

COMPUTER STUDIES: SECONDARY

Cheryl Groninger
John T. Taylor
David Hathaway

ABSTRACT: Changing the Focus of Computer Education for the College-Bound Student

Cheryl K. Groninger, North Hunterdon High School, Annandale, NJ

At North Hunterdon High School the focus of the computer education program is changing to fit today's needs of the college-bound student. When computer courses were first introduced into the curriculum, the emphasis was on programming, the accepted computer "application" of the time. However, recent developments in the field of applications software have made a wide variety of powerful software packages available for school, business, and home use. Because of the expanding possibilities in computer usage, the increasingly important role of the computer in many fields of study, and findings cited in recent literature, it was decided that there existed a need for a new type of computer course. The course would target the college-bound student and concentrate on the use of applications software.

One of the primary goals of the course is for the student to view the computer as a powerful, friendly, and effective tool for the management and retrieval of information. Members of the mathematics department at North Hunterdon designed Computer Applications to meet this goal and offered it as a semester course to college-prep students in Grades 9-12. The course curriculum has not been limited to the "Big 3"--database, word processing, and spreadsheets. An objective of the course is to include as wide a variety of software as possible in order for the student to experience the full scope of computer usage. The following topics are part of Computer Applications:

1. Computer Hardware
2. System Software
3. Databases
4. Statistics and the Computer - Sampling and Surveys
5. Word Processing
6. Programming - Investigating Symmetry with Logo
7. Spreadsheets
8. Telecommunications
9. Computer Graphics - Business and Artistic
10. The History of the Computer and Its Future - Artificial Intelligence

In as many instances as possible a concerted effort is made to relate and correlate student activities to various courses they are taking in high school or may elect in college. The activities are structured so that one or more pieces of software are used to accomplish a given task. One example is a project on the creation, administration, and interpretation of surveys. Not only are database, word processing, spreadsheet, and graphics software incorporated into the activity but also a survey tabulation program and a simulation. Other activities have included the analysis and reporting of data from a scientific experiment, charting stock market investments, and investigating the graphs of linear and quadratic equations.

Student response to the new program as measured through evaluation surveys, enrollments, and direct feedback has been very positive. Most students felt that they had greatly increased their computer knowledge and were excited about using computers in many phases of their academic and everyday life.

ABSTRACT: Magic Numbers Game: An Alternate First-Day Approach to a Computer Literacy Class

John T. Taylor, Director of Math/Science/Computer Science, Hillborough Community College, Tampa, FL

The presenter will demonstrate, by interaction with the audience, four number games written in Turbo Pascal which will introduce how a computer works using the binary number system. This presentation is used the first day of a secondary school/community college computer literacy class. The presenter will act as a human computer, processing from four to seven bits of a byte of information to identify a magic number selected by one member of the audience during each game.

Game One includes four screens of 2 x 4 arrays (eight numbers/screen). It limits the player to select any number between one and fifteen and the "Magician" to use a four-bit "nibble" to identify the number.

Game Two has five screens of 4 x 4 arrays (sixteen numbers) allowing the player to select any number between 1-31 for the game. Game Three has six screens of 4 x 8 arrays (thirty-two numbers) for the identification of any number from 1 to 63. Game Four consists of seven screens of 8 x 8 arrays (64 numbers) so that the Magician overwhelms the students with his/her "mind-reading" of any number between 1 and 127. Zero is set as the default number if the player selects "no" number on any screen.

These games introduce the concept of how switches (computer circuits) identify an "a" from an "A" from an input device, such as a keyboard by bit setting. The relationship of ASCII code to the binary number system is then shown with its use of the first seven bits of a byte.

The program also has an interactive tutorial explanation and lesson that follows the sequence of the four magic number games (15, 31, 63, 127).

One version of the program for computer with graphic cards incorporate the use of the Turbo Pascal Graphics Toolbox to allow the demonstrator the ability to project graphics screens of the game and lesson.

The first version of the program was written for MS-DOS machines. The program is available in VAX-Pascal by down loading a file from the Hillsborough Community College dial-up system. Instructions for the transfer will be available for the conference participants.

Copies of both MS-DOS versions of the program will be made available to the audience participants for their own classroom use, if they so desire.

The presenter may have Apple IIe and Macintosh versions completed by the conference date.

ABSTRACT: A Model for the Teaching of Computer Applications at the Secondary Level

David L. Hathaway, Computer Coordinator, Episcopal High School, Alexandria, VA

At West Lafayette High School in West Lafayette, Indiana, and at Episcopal High School, students are being trained to create, distribute, and procure information using the microcomputer. These are skills which will prepare them for the jobs of the '90s and the 21st century.

The course known as Computer Applications exposes students to many potential uses of the microcomputer. One of the unique characteristics is the interdisciplinary feature. The design provides the necessary structure to allow many of the traditional applications to be taught in different curricula settings by the teachers of those areas.

The model has three major divisions: business, science, and mathematics/computer science. Initially, all students meet together for an introductory unit during the first two weeks. Then the students are divided into three equal-size groups, one for each of the three major divisions. Each group spends approximately five weeks with the teacher in each area.

The model must be flexible enough to withstand the many changes the computer industry and education will undergo in the coming years. With computers there are applications. What form will they take? The model does not care. More and more elementary and/or junior high students will know about word processing, and so forth, by the time they reach high school. But will they know how to use a spreadsheet to help them analyze laboratory data in biology?

One of the most interesting aspects is how it can be used by the computer coordinator to spark interest in computers amoung his/her faculty. The above-mentioned subject areas are not sacrosanct. The social studies, English, and industrial arts departments and the library media specialist could also be given five-week periods in which to explore applications in their respective areas. Through this survey approach, educators can expose all of their children to the important information-processing skills necessary for survival in the '90s and the 21st century.

THE CURRENT STATE-OF-THE-ART IN THE EVALUATION OF EDUCATIONAL SOFTWARE: A VIEW FROM THE FEDERAL LABS

Thomas H. Saterfiel, *Chair*
Mississippi State Department of Education
Jackson, MS

ABSTRACT

The two major software evaluation activities which are currently being conducted in the federal labs in the nation are MicroSIFT, which was developed and managed by Don Holtznagel, who is Director of Technology Programs for the Northwest Regional Lab in Portland, Oregon, and SEED, which was created by a consortium of southeastern states and eventually became the Southeastern Educational Improvement Lab. In both programs, there exists a formal structure for identifying educational software packages that need to be reviewed, an established set of criteria for making judgements about the packages, and a method for disseminating information about the evaluations. Both programs have developed their procedures based on the review of the literature in the area and actual modification of their procedures based on the responses of the users of their services.

The session addresses the following points in the two projects, SEED and MicroSIFT.

1. An explanation of the history and current practice of evaluating educational software in each project.
2. An explanation of the criteria that are used in making judgements about a software package and why these criteria were established.
3. A discussion about future plans that exist within these projects for modifications in their practices or procedures.
4. A reaction about possible weaknesses or shortcomings in the area of evaluation of educational software that these projects have identified and how they feel these problems should be addressed.

Following the presentation of these two systems, there will be a formal reaction by a public school administrator who has responsibility for using software evaluation information in making decisions about what products should be selected. After the three formal presentations, the session will be open for discussion with the audience.

PRESENTERS

Don Holtznagel
Northwest Regional Lab
Portland, Oregon

Doris "Dee" Bethke
Southeastern Educational Improvement Lab
Raleigh, North Carolina

DISCUSSANT

Mike Strezeski
Garland Independent School District
Garland, Texas

SPONSOR: AERA

TEACHER CERTIFICATION: A VIEW FROM THE STATES

Tony Jongejan, *Chair*
Western Washington University
Bellingham, WA

ABSTRACT

The proliferation of computers in schools has resulted in a need for knowledgeable computer-using educators. All educators need to know how the computer can be used as an instructional tool for themselves and their students to help teach/learn class material. In addition, they need to be aware that the computer can be a personal tool for classroom preparation and management.

There is strong evidence that curriculum content must change to reflect the changing nature of the marketplace and take into consideration computer based intellectual and physical aids to productivity in our society. Thus, all educators must also be ready to adapt to these changes the computer will bring to their classes. Also, knowledgeable educators are needed who are ready to teach courses/units where the computer is the focus of study.

All these factors have resulted in a need to establish certification requirements for educators currently entering the teaching profession. This means that competencies and course requirements need to be established for the following groups:

1. All teachers;
2. Teachers by grade level or subject;
3. Computer science teachers; and
4. Computer specialists (coordinators, etc.).

In the category of "all teachers," there is a core of knowledge and skills independent of grade/subject area with which all teachers should be competent. These are based on a three-part model. That is, all future teachers should be provided with a learning experience that deals with:

1. The computer as a system;
2. Computer applications in education; and
3. The computer as a tool for personal use.

After the introductory experience for all education students, additional content should be integrated into all required and optional courses in the teacher preparation program. These experiences should match the specific program needs of the individual students followed by knowledge and skills specific to individual grade level/content area.

Educators who are going to teach courses/units where the computer is the focus of study need additional formal training. Computer science teachers need formal training in computer science. Recommendations for this training have been specified by the Association for Computing Machinery (ACM) and endorsed by the International Council for Computers in Education (ICCE) Teacher Education committee.

Finally, computer specialists (coordinators, etc.) need different skills, knowledge, and perspective. They need supervision, curriculum development, and administrative skills in addition to a broad forward-looking perspective on the role of computers in their school and/or district.

Individual states are beginning to address the issue of teacher certification, and some states have legislation in progress. This panel will discuss the process of establishing certification requirements and look at the progress of certification in certain states and nationwide.

In addition, it should provide information for state committees to use in developing their own teacher certification requirements and for Schools of Education to use in developing their computer education curriculum.

PANELISTS

David Brittain
Florida State Department of Education
Tallahassee, FL

Bobby Goodson
Sunnyvale, CA

Don Knezek
Texas Education Agency
Austin, TX

SPONSOR: ICCE

ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: BACKGROUND PAPERS II

John Lawson, *Chair*
Chair, ACM/SIGCUE
Towson State University
Baltimore, MD

ABSTRACTS

One of the major concerns of computer-using school districts is how to train all of their teachers to be effective consumers of technology. Across the country millions of dollars are being spent retraining teachers to integrate computing into the curriculum. Yet with all this concern over the need for computer-using teachers, and over the cost of retraining current teachers, teachers, just now emerging from preservice programs are little more prepared than their existing colleagues.

The purpose of this project is to develop an interim set of guidelines for the addition of one solid course on the use of computers to existing preservice education programs. As a part of the development, papers were solicited from individuals to serve a background papers for the course development. These papers were reviewed at an invited conference held in January at Teachers College. The papers were then revised and published in the Spring '88 issue of SIGCUE Outlook. This is the second of two sessions at NECC to discuss the papers. Background papers to be discussed in this session are:

You Teach What is Global and Leave the Specifics to Me

by Sheila Cory

School districts do not have adequate resources to properly train teachers to full utilize computer and emerging technologies in their instructional program. Therefore, schools of education should offer a required preservice course in computer education that focuses on the global (not specific) issues. The global issues are those that are relatively constant across time, across different kinds of hardware and operating systems, across different software titles, across grade levels, and across different content areas. These global issues are discussed in five major categories including the time continuum of educational computing, the role of the computer in education, social issues relating to computers in education, issues relating to hardware, and issues relating to software. Such a preservice course, the computer-related counterpart to a course in educational foundations, would provide a framework for understanding and adapting to the whole array of possible situations a teacher might encounter on the job.

The Computer as Microworld and Microworld Maker; A rationale and Plan for the Inclusion of Logo in an Introductory, Preservice Course on Educational Computing

by Daniel H. Watt

This paper makes a case for the inclusion of Logo and Logo-based microworlds as part of an introductory course on educational computing for preservice teachers. Logo is a interactive, exploratory, mathematical problem-solving environment as well as a flexible, powerful, easy-to-learn programming language. As such, it is ideal for illustrating three important ways in which computers can be used to enhance learning. First, Logo is a domain in which key ideas of elementary computer science can be readily introduced. Second, Turtle Geometry, the mathematical learning environment built into Logo, illustrates some of the ways in which computer-based microworlds can foster exploratory learning of important mathematical ideas. Third, Logo can be used by teachers as a programming language in which to develop exploratory learning activities connected to specific curriculum objectives. This paper offers a nine-hour sequence of Logo activities designed to introduce these ideas and encourage prospective teachers to learn more about Logo at a later date.

Experiential Learning: Elements to Consider in Designing a Preservice Course in Educational Computing

by Molly Watt

This paper describes a method of teaching educational computing which would model effective teaching strategies which can be appropriately adapted for school classrooms. The way teachers learn a subject is usually how they teach it. Since educational computing is new to most preservice teachers, the paper makes and an argument for making teaching and learning with the computer part of the focus of the course.

The paper assumes that learning about educational uses of computers will require teachers of preservice teachers to use a variety of methods. Lecture and demonstration may make up about a third of the student's class time in order to provide some overview of the historic perspective and the special educational possibilities the computer affords. Another third of the total class time would be hands-on computer time with the teacher of the course modeling effective student interventions, challenges and ways to support problem-solving, project development, collaborative work, teacher conferencing and debugging in a computer lab setting. The third third of class time would be the opportunity to participate in whole class, small group and partner discussions reflecting on or interpreting the educational value of various activities, their own learning process and the teaching strategies and structures used.

The author assumes that by the end of a course taught in this way, preservice teachers will have many teaching strategies and the confidence to use computers to support their particular curriculum in a manner appropriate to the learners, the curriculum and the computer. Preservice teachers will have a well articulated philosophy of educational computing. Plus they will know a great deal about actually using the computer as a tool for their own work.

Using Computers to Enhance the Mathematics Classroom

by Sharon Whitton Ayers
and Gary Bitter

This article presents a model for integrating the computer into a middle grades mathematics classroom. It explores how computers may be used to motivate, teach and clarify various mathematical concepts. Effective applications of pre-packaged software, databases, spreadsheets and graphics packages are illustrated. A rationale and methodology for including introductory programming as a tool for the teaching of problem solving and certain mathematical concepts are included. The content and methods described in this article could be used as a unit in a single preservice course in educational computing or as a course for middle grades teachers.

Addressing the Needs of the Handicapped Child

by Gerry Segal and Vincent Marciano

The article focuses on the use of adaptive technology for the disabled student and on how the issues mentioned below should be addressed in a course for all future teachers.

Disabilities are discussed in 3 categories: mobility impairments, sensory impairment and learning disabilities. A framework of adaptations for each group has been established, allowing the educator to think about the computer hardware options available based on compensatory diagnosis.

It brings to the surface the various issues that preservice education programs should be made aware of. Such as, the classroom technologies that hinder the disabled student; problems and limitations of the solution strategies; future potentials and problems; and the Federal and State laws addressing training of the disabled student.

It goes further to suggest and discuss strategies that preservice schools might implement. Finally, it reviews the sources that are available to assist educators in selecting software and hardware for the handicapped.

College-wide Administrative Issues in Implementing a Preservice Course in Computing in Education

by Dede Heidt and Jim Poirot

While the problem of teacher retraining in computing has received considerable attention over the past several years, the need to serve teachers who are now being prepared to go into the classroom, that is, the preservice teacher, is still a pressing one. It is imperative that the graduates of our universities receive an adequate background in the utilization of the computer in the classroom prior to their graduation.

It is unfortunate, however, that even though computing technology is pervasive in the pre-college market, the training of elementary school teachers at the university level in computer utilization is lagging far behind. Why is the university system failing in its efforts to prepare teachers for computing technology? That quaestion and other issues will be addressed in this presentation.

Administrative and Fiscal Issues Coping with the Real World

by Barbara Feddem

Although there has been a proliferation in the number of computers utilized in the schools, and the quality of software is improving, teachers are still facing difficulties in integrating computer technology into their classroom curricula. Few districts have comprehensive plans, and quality teacher education lags far behind hardware and software acquisition. Administrative policies and financial restraints are at worse counter productive and at best discouraging.

There are many strategies a teacher may employ to circumvent or alleviate many of the problems. Hardware and software incompatibility, software selection, copyright violations, administrative negativism, building politics, staff development, and financial restraints are discussed, and suggestions for coping with them are proposed. New teachers need to be aware of the stumbling blocks and the alternative for working within the system and initiating change whenever possible.

PANELISTS

Gary Bitter
Arizona State University
Tempe, AZ

Ruthie Blankenbaker
Park Tudor School
Indianapolis, IN

Barbara Feddern
The Center for Learning Technologies
Long Island, NY

Daniel Watt
Center for Learning Technology
Newton, MA

SPONSOR: SIGCUE

A MODEL FOR TEACHING THINKING SKILLS USING ONE COMPUTER, ONE CLASS, ONE TEACHER AND PROBLEM-SOLVING SOFTWARE

Robert M. van Deusen
Grant Wood Area Education Agency
Cedar Rapids, IA 52404

ABSTRACT

Can questioning strategies help teachers model thinking processes in a nonevaluative, whole class environment? Does problem-solving software create a sufficiently rich environment for children to fabricate their own understandings of sequence, hypothesis testing, inference, and data organization without teacher direction? Does the classroom environment (accepting vs. evaluative) determine the level of risk that students will take?

A teaching/learning model which examines these and other questions which change the nature of learning, the conception of the learner, and the role of the teacher will be demonstrated and discussed. Whole class teaching with one computer and a large color monitor can be a vehicle for **process** learning. One lesson, consistent with the model, will be taught to help you see how you can help students process information, use logical evidence to defend an opinion, hypothesize, infer, plan, analyze, verbalize, collaborate, and explore.

The teacher should question students to cause them to examine their own thinking, whether they are right or wrong. This kind of nonevaluative questioning is an essential component of an effective lesson involving higher order thinking skills. The teacher should also encourage students to pursue blind alleys, dead-end streets, and other divergent paths because the critical thinking of students can be engendered through such a process. Nonevaluative responses from the teacher provide an environment conducive to taking risks. Questioning strategies and nonevaluative responses combine to produce a new learning genre for students, one that is transferable to noncomputer lessons where a concept or process is being taught.

When teaching a concept, higher order thinking skill, problem-solving strategy, or other process-oriented lesson, the teacher needs to avoid evaluative responses. Evaluative responses cause students to try to guess what the teacher wants. Nonevaluative responses allow students to explore their thinking and try ideas that would never be risked in a convergent content-oriented lesson. Evaluative responses include praise, criticism, and constructive suggestions. Nonevaluative responses are characterized by accepting, waiting, and then further questioning. "What do you think?" [Teacher pauses after the student's response before asking another question.] "Why do you think that?" [Teacher pauses after the student's response before asking another question.] "How did you figure that out?" Nonevaluative responses allow the teacher to become the facilitator, the question-asker, but not the answer-giver. Nonevaluative responses put the responsibility for thinking and learning on the students.

Many pieces of excellent problem-solving software can be trivialized by solving the problem without any additional constraints placed on the students by the teacher. In the hands of a skilled teacher, the task can be made much more challenging by dividing the lesson into three parts. Begin by asking the students to first determine the rules (instructions) by use of induction instead of reading and applying the instructions on the screen. Second, determine an algorithm to solve the problem in the fewest possible number of moves. Third, test the algorithm. Children working at the computer **do not, can not** and **will not** ask these questions of themselves. The role of the teacher in the process of teaching higher order thinking skills is crucial.

DATABASES: TEACHING THINKING SKILLS IN CONTENT AREAS

Jean Donham
Mary Jo Langhorne
Denise Rehmke

Library Media Program
Iowa City Community Schools
Iowa City, IA 52240

ABSTRACT

The computerized database, frequently used as a management tool, has a place in the classroom as an instructional tool.

There are three distinct reasons for introducing students to databases and designing instruction which makes use of them. First, by learning about databases and developing skill in manipulating a database, students are attaining a vital "information age" concept and skill. Students today are entering into a technology driven, information-rich society where skill at using technology to access information is essential. Any awareness, insight or experience our students can gain about databases—the basic unit for most information management in society—will be an edge for them.

A second reason for using the database as an instructional tool is that it provides a means for teaching higher order thinking skills such as classifying, comparing, contrasting, drawing inferences, hypothesizing, generalizing and using Boolean logic. Students can be guided by the questioning strategy of the teacher to use various higher order thinking skills while manipulating databases.

Beyond this, the database can also serve as an instructional vehicle for imparting specific content. This, then, is the third reason for teaching with a database. Students need to understand what a database is, how to develop a search strategy, and how to manipulate a database to perform a search. The classroom teacher can choose a database as a tool for teaching specific information. By interrogating and manipulating a database, students can make generalizations, then proceed to validate or negate them.

This presentation suggests a series of lessons designed to introduce students to the concept of database and the sorting and searching processes involved in manipulating a database. The presenters will summarize the lessons, model several of the activities and show videotaped segments of the lessons in progress with students.

In the first lesson, the students learn key database terms and how a computer sorts information. In the next lessons, students practice sorting and are introduced to the Boolean concepts of **AND** and **OR** through the use of attribute manipulatives and problem solving software. The students are guided in developing strategies for solving problems using a database. The questioning strategies used by the teacher to focus on thinking skills will be modeled and discussed.

The lessons themselves can be integrated into various curricular areas. Possibilities suggested include mathematics where Boolean logic sorting lessons are appropriate or library media classes where database use is approached as another information accessing skill. Database use beyond the lessons can be integrated into such content areas as social studies or science. A database activity used in a life-science class will be described.

THE USE OF INTERACTIVE VIDEODISCS IN SCIENCE AND MATH INSTRUCTION

John Woodward
College of Education
University of Oregon
Eugene, OR

ABSTRACT

As an instructional tool, videodisc technology offers many important features that place it well beyond what can be found on today's microcomputers. Furthermore, it allows an interactive format which is not possible with conventional audio visual materials (e.g., film strips, films, video cassettes). Vivid visual demonstrations can be used to present concepts, thus rendering them easily understood. Computer graphics, sound effects, highlights, and other techniques visually and auditorily are effective techniques for stimulating students. The medium allows instructional designers to incorporate detailed displays of concepts (e.g., the effect of increases in temperature on the velocity of molecules, ionic bonding, convectional movements in the earth's mantle) and insert questions which are asked by a narrator. It is well suited to content areas that require demonstrations which are difficult or expensive to conduct in classroom situations. Finally, the videodisc is a particularly powerful technology for academically low achieving students because difficult concepts can be presented visually, and narration that accompanies these demonstrations reduces the need for students to read lengthy text explanations.

The focus of this presentation is on several areas of recent research and development in videodisc instruction for math and science. Two experimental studies have involved comparisons of particular videodisc programs in math with their basal counterparts. In this research, basal curriculum was refined and presented in an effective and consistent manner. Both studies examined specific instructional design variables, and results from each study favored the videodisc program. In the first study, the effect of the range and type of examples used to present fundamental concepts in fractions was the key variable. In the second study, researchers investigated the effect of different strategies for teaching ratio word problems.

A complementary area of research has involved more naturalistic settings. If nothing else, the research on educational change and innovation shows that it is crucial to assess and evaluate how well a new curriculum, new technology, or new model of teaching is implemented on a day to day basis by classroom teachers. While educational innovations may be conceptually elegant and work well in experimental settings, they may lead to less than satisfactory results when implemented in a serious, consistent fashion in public schools. This part of the presentation will focus on two large scale evaluations of videodisc instruction and its use in a range of classroom settings, from high achieving to resource room instruction. Results of these evaluations again show that the videodisc instruction was superior to traditional curriculum and that it was favorably received by classroom teachers.

The final portion of this presentation will address the use of videodisc instruction as a key component in an advanced CAI program currently under development for the Office of Special Education Programs. Designers are incorporating the videodisc as the medium for presenting initial concepts and information. This technology was chosen not only because it is an optimal means for demonstrating complicated concepts such as convection and dynamic pressure, but because adequate visual presentations could not be accomplished on current microcomputers. By assigning the videodisc to the most demanding aspects of instruction (i.e., the initial presentations and explanations), a microcomputer is used to present small scale simulations and track student performace over time. This selective use of technologies is but one way that videodiscs can be incorporated into advanced CAI systems.

THE ROLE OF COMPUTERS IN THE EVOLVING SCIENCE CLASSROOM

Heidi Imhof, Richard Adams and Robin McGrew

ABSTRACT

This presentation will demonstrate several ways in which computers may be used as tools to make science lessons more dynamic. Ideas for earth, life and physical science classrooms will be shared, appropriate for middle through high school level. With a focus on relevant, primary-source data collection, the presenters will explore the use of microcomputer-based labware (MBL), database/spreadsheet applications, and the expanding role of student research projects in science classrooms.

The equation, *distance = rate * time*, is the theme of both an MBL and spreadsheet lesson. The MBL approach focuses on the advantage of working with real time data displays that are extremely interactive and provide immediate feedback to students. In the spreadsheet lesson, students explore the mathematical relationship of *distance = rate * time* with some unusual examples, chosen to illustrate the somewhat tentative nature of science.

Databases can provide stimulating investigative experiences in existing lessons. Thoughtful database template design, critical to the *shaping* of data, enhances the exploration of new research questions. Developing research design skills, like primary source data collection and database template design, may enable students to understand the process of hypothesis generation, and science itself, at a more fundamental level.

Teachers who use computers as tools in their science classrooms may empower students to pursue their own ideas and ask their own research questions, a direction that holds much promise for science classrooms of the future.

DESIGNING AN INTELLIGENT CURRICULUM PLANNER

Karen McGraw and Barbara Brown[1]
Texas Instruments Defense Systems and Electronics
Artificial Intelligence Laboratory
P.O. Box 660246, MS 3645
Dallas, TX 75266

ABSTRACT

Recent advances in the field of Intelligent Instructional Systems, a subfield of Artificial Intelligence, may provide enhancements to the role of the computer as an educational management tool. The computer has long been touted as both a tool and tutor in the education arena. As a tool, it can assist in educational management functions such as compiling test banks, statistics, and generating student reports. As a tutor, it can assist by presenting instruction or an entire environment for individualized instruction. In its combined roles of tutor and tool, it provides an integrated approach to Computer-Based Training. The improvements of the last decade in the field of Computer-Based Training notwithstanding, the computer's usefulness as a job aid for scheduling or planning curricula has been limited. This anecdotal paper describes efforts in the design and initial implementation of an Intelligent Computer Managed Instruction tool for curriculum planning. Specifically, we describe the design for an intelligent curriculum planner that consists of a learner model, individual learner profiles, skills taxonomies, resource databases, and a plan generator.

INTRODUCTION

The purpose of this paper is to present a brief introduction to Intelligent Computer Managed Instruction (ICMI) and to describe anecdotal efforts in the design of an intelligent curriculum planner for a limited domain. First, we compare the major components and capabilities of CMI and ICMI to provide background concepts for the field. Next, we discuss the problem selection and overall approach to the development of a curriculum planner. In the subsequent section we detail the major development phases and describe current functionality. We conclude with a brief discussion of some of the major unresolved issues and considerations for further development.

The Problem

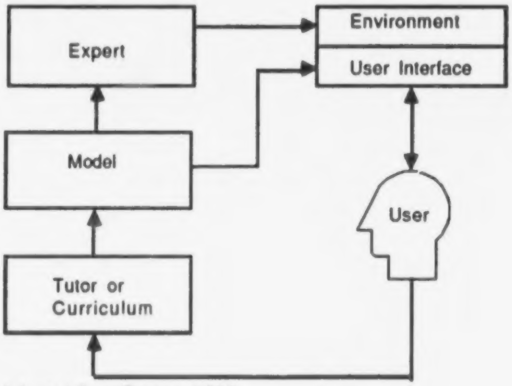
Computers have been used as instructional delivery and management systems for decades. Early research by Suppes, Jerman, and Brian (1968) suggested that the addition of brief drill and practice with a computer to regular basic skills *instruction* could significantly improve student achievement. The computer's usefulness as a tool to *manage* instruction was realized by research (Suppes, Fletcher, and Zanotti, 1975, 1976) that revealed the type of detailed performance data that could be collected from student learners during their work with computer-based instruction. Major computer-based development programs which later became commercially available, including PLATO and TICCIT, extended the applicability of computer-based instruction. The paradigms and techniques used in the development of computer-assisted and computer-managed programs for these systems were transferred to programs that were later developed for microcomputer-based delivery. However, computer-based instruction and management has often been criticized for its use of limited paradigms (i.e., drill and practice, programmed instruction) and lack of flexibility.

Applying AI Toward A Solution

The primary goal of intelligent computer assisted instruction (ICAI) and intelligent computer managed instruction (ICMI) is to develop more powerful and accurate adaptive instructional systems (Park, Perez, and Seidel, 1987) by applying selected Artificial Intelligence (AI) principles and techniques. These techniques include developing:

- representations of domain knowledge (e.g., the "expert" component)
- models of learner behavior (e.g., the "learner model"), and
- a method for isolating what a student knows from what a student needs to know (e.g., the "overlay" method).

Additionally, systems include a component described as the "environment" that allows users to isolate the mistakes a learner is making during learning episodes. To allow users to interact with the system, both ICAI and ICMI theory advocates the addition of an easy-to-use user interface (Figure 1).



Adapted from Burton, 1986

Figure 1. Typical System Components

Previous efforts to apply artificial intelligence techniques to instruction have been limited almost entirely to research domains and to ICAI as opposed to ICMI. The purpose of this paper is to describe early efforts to apply artificial intelligence to the problem of intelligently "managing" or planning a curriculum in the domain of writing. This "proof of concept" project was undertaken to investigate the feasibility of developing and using an ICMI program in this domain in an in-house training environment.

INTELLIGENT COMPUTER MANAGED INSTRUCTION

ICMI can be compared to both its larger field, ICAI, and its more traditional ancestor, Computer Managed Instruction (CMI). While early efforts by Suppes and others in the field of Intelligent Computer Assisted Instruction have provided a strong base of accepted development models, components, and techniques, ICMI has received far less individual attention. The following section presents some comparisons of CMI and ICMI and lays the groundwork for the discussion of efforts in developing an ICMI tool to manage, or plan a curriculum.

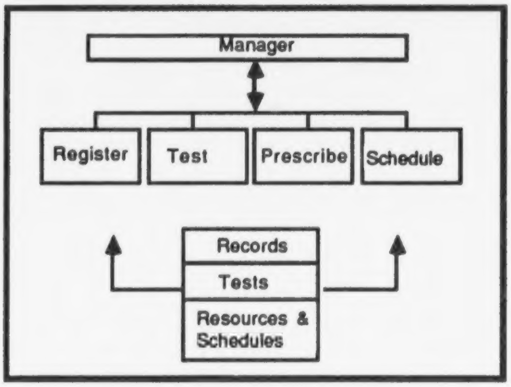
Comparisons: CMI and ICMI

Although both deal with the storage of student educational-related information and the management of computer-based instruction, there are basic differences in CMI and ICMI. While ICMI typically embodies at least the same information as CMI, capabilities for using that information, extending that information, and decentralizing that information are enhanced in ICMI.

Computer Managed Instruction Computer-management of instruction (Baker, 1978) was often an embedded component in traditional computer-based instruction programs. The core of traditional CMI was commonly a computer-based testing component. Testing was the means by which training achievement was measured and thus, it was the key element upon which other components depended (Kearsley, 1983). CMI used the computer to manage student progress and teaching resources and to provide a range of information related to learner interaction with the program or a specific lesson. Teachers reviewed the information provided by the management component or CMI system to gain information that was used in adjusting or planning follow-on, enrichment, or remedial work. Typical information provided by traditional CMI systems included the following:

Student Identification
Date of Interactions
Activity Accessed
Time on Task
Ratio Correct to Incorrect
Class Ranking
Materials Used
Descriptive Statistics on Items
Total Percentage Correct
Specific Items Correct/Incorrect

Other information that was often available within a CMI system included a component in which students could register, a component that managed the user-system interaction, and a data base of test items and student profile information (Figure 2).



Adapted from Kearsley, 1983

Figure 2. Standard Components of Computer Managed Instruction

Intelligent Computer Managed Instruction. In comparison to traditional CMI, Intelligent Computer Managed Instruction extends the power and usefulness of the CMI components. For example, in addition to the information provided by CMI systems, ICMI systems can provide the following types of information:

- Comparisons of individual learner profiles with a learner model to identify necessary, but absent skills
- Hierarchical representations of domain skills, indicating skill prerequisites and primitive skills (e.g., those for which no prerequisites are presumed)
- Individual learner profiles, which include information (e.g., needed skills, scheduling constraints, preferred learning modality, previous attempts using a resource) that is used in the planning and presentation of material to that learner
- A weighted data base of resources (e.g., courses, tapes) to facilitate the acquisition of specific skills
- Dynamic updating of learner profiles based on resources attempted to gain skills and the results of that process
- Generation of learning plans or schedules based on a learner's profile and weightings of learning resources under consideration
- Management of conflicts during plan generation.

Many applications and adaptations of ICMI are possible. In some cases, ICMI systems may be a component within an ICAI system. In others, management of some set of elements may be the primary system goal. Thus, the architectures in which the systems are embedded might be vastly different, making it difficult to define typical ICMI system components. For example, in the ICMI system discussed later in this paper, the overall goal for functionality was for it to plan the curriculum for individual learners. In such a system the following major components are necessary: Learner Model, Individual Learner Profiles, Skills Taxonomy, Resource Database, and the Plan Generator.

A CURRICULUM PLANNER

The purpose of the project we describe was to develop and test the feasibility of a generic curriculum planner that would be capable of creating an individualized training plan. The end product of the project was to be a curriculum scheduler, which would suggest a sequence of courses and materials to maximize the effectiveness of training by building on previously-learned skills. In addition to design-specific or topic-selection considerations, a major constraint was that this planner embody a training topic relevant to the needs of our corporation.

Selecting the Domain

A skill that is frequently called upon within either an educational or corporate setting is the ability to communicate effectively in writing. In large corporations the ability to write well translates into proposals that win contracts, brochures that reflect company strengths, and effective interpersonal (and inter-group) communications. For those with limited writing skills, internal education programs may be called upon to enhance writing ability in general. Furthermore, even when an individual writes effectively, instruction in the "art" of writing effective proposals is beneficial. Because of its applicability to corporate needs and its suitability for the proof of concept phase of the ICMI project, proposal writing was selected as the initial problem domain.

The Overall Approach

Writing a proposal that wins contracts or project support involves multiple knowledge areas. For development purposes, the proposal writing domain was divided into the following eight modules:

Request for Proposal Response
Organizing the Proposal Writing Effort
Proposal Writing Style
Using Graphics and Illustrations
Scheduling
Using Computer-aided Text Design Tools
Storyboarding
Assessing Company Strengths

Initially, personnel should have the ability to respond to the "request for proposal" that a potential customer or foundation might publicize. As work on the proposal itself begins, personnel assigned to the project should be able to organize the proposal and use an effective writing style to draft the proposal. Extending the message to supporting figures, personnel should also know how to develop and use graphics and illustrations for maximum effect (or who to call for assistance in doing so). Other areas in which instruction may be desired include scheduling, storyboarding, using computer aided text design tools (e.g., Pagemaker^{TM}), and assessing corporate strengths and weaknesses.

Within the *proof of concept* phase of development, the primary task was to select and develop only one of these modules. For demonstration purposes, the chosen module had to meet the following requirements. First, designers should be able to decompose the module into hierarchically-sequenced sub-skills. Second, the module must have a well-defined knowledge base. Third, the domain the module reflected must be one for which a variety of training resources were already in existence to allow individualization in the schedule and to reduce the time required for knowledge acquisition. Applying these constraints, the Writing Style module was selected for initial development efforts.

DEVELOPING THE WRITING CURRICULUM PLANNER

Previous authors (Prerau, 1985; Buchanan, 1983) have suggested a sequence of activities for the development of programs that use artificial intelligence techniques. These activities generally include selecting the domain and a sample problem, selecting the appropriate implementation tool, acquiring knowledge, and representing the knowledge to produce the program. The previous section discussed considerations for domain selection. The following sections (1) describe efforts in tool selection and (2) knowledge acquisition/implementation and (3) report on current functionality.

Selecting the Tool

After selecting the training topic, the development team reviewed a number of development tools for use in the project. The selected tool would be required to run on the Texas Instruments Explorer^{TM2} because of their ready availability with the company. Additionally, it was determined that an "intelligent" tool would be advantageous to solving our problem as it would provide the flexibility to generalize to other topics at a later date and the ability to reason about information in the developing knowledge base. Eventually the *Knowledge Engineering Environment* (KEE^{TM}) from Intellicorp was selected. Selection was based on the fact that KEE allows access to a large knowledge base and is capable of reasoning about that knowledge base through rules. Additionally, KEE has a facility called inheritance, which assists in the representation of hierarchical information. Finally, KEE allows developers to work with multiple worlds, through which more than one plan can be devised for the comparison in order to select the plan that best meets identified goals.

Knowledge Acquisition

Knowledge acquisition entails acquiring and structuring knowledge that is translated into the rules that become the heart of an expert system (McGraw & Briggs, 1988). Selected knowledge is identified, conceptualized and acquired either from human experts, existing print material, or both. Initial knowledge acquisition efforts occur during what Buchanan (et.al., 1983) terms the Identification stage. Initial knowledge sources for this stage of the intelligent planner project included the following:

- College English Curricula
- Technical Writing Texts
- Proposal Writing Texts
- Interviews with "Experts" in Proposal Preparation.

The next phases of knowledge acquisition involve conceptualization and analysis tasks. During these phases, the sub-domain of "writing style" was analyzed to identify the skills that the system would be required to represent. Target skills that were identified during this phase ranged from basic grammar to text formatting and presentation style. Once identified, each skill was decomposed into a set of baseline (e.g., enabling) skills and the resources that were available to teach each of the baseline skills were identified.

Creating the Skills Taxonomy. During this conceptualization and analysis phase, the identified knowledge is structured or grouped to enable the knowledge engineer to understand critical prerequisites and interrelationships and to identify gaps in the set of identified skills. To accomplish this task each identified skill was ranked according to its perceived level of "difficulty." Skills that were considered to be prerequisites to the successful acquisition of other skills were termed baseline skills. Using KEE, a taxonomy or branching hierarchy of skills was then developed. Figure 3 displays a portion of the skills taxonomy. Note that prerequisite skills are considered to be "child" skills of more difficult or complex "parent" skills. This structure allows the addition of training topics other than writing (e.g., using graphics), which can be decomposed, embedded into the structure, and considered when generating a training schedule for the suggested skill acquisition sequence.

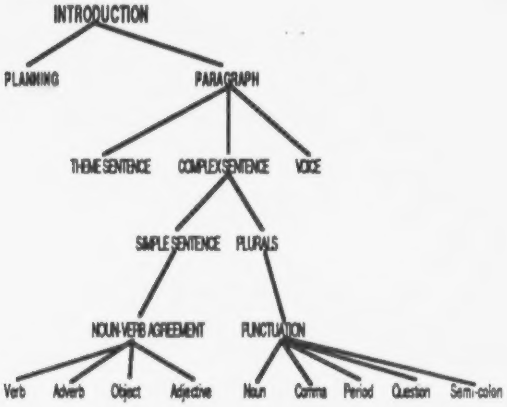


Figure 3. Partial Skills Taxonomy

Rule Development. The planner consists of both declarative (e.g., "what") and procedural (e.g., "how") rules. Declarative rules included information such as prerequisite skills for an objective or course or available materials to help in meeting an objective. Figure 4 presents an example rule that embodies declarative information. Procedural rules included information such as the materials that were most effective if the learner input a specific learning mode preference or time constraints, or if cost is a major factor.

Figure 4. Pseudocode for a Declarative Rule

Although the previous phase of knowledge acquisition focused on the identification of skills and available resources, the translation of resource selection heuristics (e.g., "rules of thumb") into rules required that the development of weightings that would indicate which resources were most (and least) desirable. Ranking criteria that were identified and evaluated for each resources included the following:

- Effective Style of Presentation
- Number of Target Skills Presented or "Taught"
- Reflects the Preferred Learner Modality
- Completion Time
- Cost.

Functionality

The primary goal for this stage of the development of the curriculum planner was to provide a suggested sequence of courses and materials based on a learner's input regarding previously learned skills, preferred learning style, cost, and method of presentation.

Developing the Learner Model. To produce a plan within the system, the learner is prompted for information about entry level skills and preferred learning mode (e.g., visual, audio-visual, or tutorial). Entry-level skills could be determined either by pre-testing or by individual report, depending on the subject matter. Similarly, preference for a specific learning modality could be derived from a test or by individual report; for our initial purposes, the learner input this information based solely on individual choice (i.e., self-report). Timing or cost factors may also be inferred.

After the learner has input required information about current skills and favored learning mode, the planner generates a list of skills that the learner needs. This list is generated using the "overlay" method, which compares the nodes on the Learner Model to the nodes that represent the current learner's acquired skills. Matching nodes are pruned away from the current learner's "taxonomy," leaving only the skills which need to be acquired.

Generating the Skills Taxonomy. Once a list of skills on which the learner is to be trained has been generated, the planner sequences these skills based on the learning taxonomy or hierarchy. The Skills Taxonomy reflects a branching hierarchy of writing style skills. The purpose of this process is to ensure that no skill is scheduled for training before the learner has mastered the prerequisite skills that will contribute to the effective transfer of the selected skill.

Generating the Plan. After the target skills a learner needs have been identified and sequenced, the planner weighs a set of constraints that are used in developing the suggested curriculum "schedule." Resources (e.g., courses, videotapes, texts) that support or "teach" the target skills are ranked based on weighting factors that were identified during knowledge acquisition. A highly ranked resource, for example, should indicate that the resource reflects an effective style of presentation, presents a high percentage of the target skills, matches the learner's modality preference, has a reasonable completion time, and is of moderate cost.

To develop the preferred list of resources that teach the target skill, a preliminary list of all resources that teach the target skill is generated. Next, the list is evaluated based on the weightings described previously. Any resource that violates time constraints is excluded immediately and the remaining resources are ranked. Based on the most highly ranked resources, a schedule or plan for skill development (e.g., classes, self-study texts) is generated and analyzed for internal time conflicts. For example, if two resources require the same time slot, or teach highly similar skills, the lower-ranked resources is excluded and a revised plan is prepared.

SUMMARY

This paper has described anecdotal efforts in the design and implementation of an Intelligent Computer Managed Instruction system that is to function as a curriculum planner in a subset of the written communication domain. Key issues that were confronted during this development phase included

- the representation of domain skills in a taxonomy,
- the ranking of domain skills as to difficulty,
- the identification and weighting of learning resources to enable the acquisition of specific domain skills,
- the design of the learner model, against which individual learner profiles can be compared, and
- the use of KEE as a development tool and user interface.

At this stage, evaluation for success is based on the *feasibility* of such an approach rather than on its specific implementation. Therefore, there are a number of questions concerning implementation that remain unanswered and provide fertile soil for further research efforts. For example, at this point in the development of the proof of concept planner, we consider the completion of a learning resource to automatically indicate the successful acquisition of the skills it is professed to "teach." We realize, of course, that this idealistic view is overly simplistic. Future research efforts might include the creation of a more traditional testing component to present and embed the results from pre-tests. Similarly, we assume that learners can accurately report on their preferred learning mode. While this may be possible, *preferred* mode may not necessarily be synonymous with the most effective mode for a particular learner or skill. Again, incorporating simple survey-style assessments in the pre-testing component can allow developers to build the system to infer effective learning modes rather than rely on self-reports. Then the system can use this information when selecting resources to enable planned instruction. Similarly, heuristics can aid in selecting a presentation style that is known to enhance a specific skills.

Finally, further development efforts are needed to incorporate a more appropriate user interface for a system such as this. Key to this design effort is identifying target users (corporate trainers versus individual learners). Once completed, enhancements need to be made to the manner in which a skills taxonomy and resource description is displayed and the amount of freedom a user is to have in modifying portions of the knowledge base.

REFERENCES

Baker, F. *Computer-managed instruction: Theory and practice.* 1978, Englewood Cliffs, NJ: Prentice-Hall.

Buchanan, R; D. Barstow; R. Bechtal, J. Bennett, W. Clancey, C. Kulikowski, T. Mitchell, and D. Waterman. Constructing an expert system. In Hayes-Roth, F., Waterman, and D. Lenat, *Building Expert Systems*, Addison-Wesley: Reading, MA, 1983.

Burton, R. The environment module of intelligent tutoring systems. Minutes of the Research Planning Forum for Intelligent Tutoring Systems, 1986, San Antonio, TX.

Kearsley, G. *Computer-Based Training: A Guide to Selection and Implementation.* Addison-Wesley: Reading, MA, 1983.

McGraw, K. and K. Briggs. *The Knowledge Acquisition Handbook.* 1988, Englewood Cliffs, NJ: Prentice-Hall.

Park, O., R. Perez, and R. Seidel. Intelligent CAI: Old Wine in New Bottles, or a New Vintage? In Kearsley, G. (Ed.) *Artificial Intelligence and Instruction: Applications and Methods.* Addison-Wesley: Reading,

Prerau, D.Selection of an appropriate domain for an expert system. *AI Magazine*, 1895, 4, 2, p. 26-30.

Suppes,P., Jerman, M. and Brian, D. *Computer-assisted instruction: The 1965-66 Stanford arithmetic program.* New York: Academic Press.

Suppes, P. Fletcher, J. and Zanotti, M. Models of individual trajectories in computer-assisted instruction for deaf students. *Journal of Educational Psychology*, 1976, 68, 117-127.

Suppes, P., Fletcher, J. and Zanotti, M. Performance models of American Indian students on computer-assisted instruction in elementary mathematics. *Instructional Science*, 1975, 4, 303-313.

---

[1] Dr. Karen McGraw currently heads Cognitive Technologies, P.O. Box 15598, Alexandria, VA 22309.

[2] Trademark of Texas Instruments, Inc.

MAKING CONNECTIONS: *LOGOWRITER*, TELECOMMUNICATIONS, AND SOCIAL STUDIES

Leon Foster, Computer Coordinator
Bay Trail School, Penfield, New York

Project Teachers
Barbara Card
Beverly House
Eugene Benoit
Phillip Osterling
Carol Hartman
David Lynch

ABSTRACT

The computer is used as a tool for students in the understanding of the relationships of information and skills learned in varying environments. Similarly, the computer is a tool in the synthesis of those skills when used in the context of creative projects. Using telecommunications and *LogoWriter*, students at Bay Trail Middle School in Penfield, New York in a cooperative project in which they conduct research for a series of questions proposed by their Social Studies teachers via an on-line information service, write a position statement, and, using *LogoWriter*, program a graphic and text project based on that research.

This paper discusses the teacher planning, defines the nature and philosophy of the project, describes the instructional unit, and presents our experiences and evaluations of the project in the context of computer education at Bay Trail School.

BACKGROUND

Computer education in Penfield begins in grade 2 and continues through high school with a strong focus on the use of the computer as a tool. Logo is introduced at grade two and continues throughout elementary school. Word processing and data base skills are introduced at grades four and five respectively.

At Bay Trail Middle School, computer education includes a daily ten-week course at grade seven and at grade eight.

Seventh grade students use the computer for word processing, for problem solving - using *LogoWriter* - and for telecommunications. Using *AppleWorks*, student writing emphasizes writing and editing skills through journal entries and letter writing. In seventh grade the telecommunications unit is based in bulletin board services and E-Mail. Electronic mail is presently accomplished through the use of E-Mail simulation programs, including *The Electronic Village* and *Electronic Mailbag* and through bulletin board services available on *Compuserve.*

Further application of word processing skills, data base management, continued projects in Logo, and in telecommunications comprise the computer education course for eighth grade students. The telecommunications unit focuses upon the use of the computer as a research tool - using the on-line information services of *Compuserve* - especially Grolier's Academic American Encyclopedia and IQUEST.

PROJECT ORIGINS

The *LogoWriter*/Social Studies /Telecommunications project arose from ongoing curriculum development during the summer of 1987 as accomplished by our team of computer teachers. Computer education teachers come from a variety of disciplines within the school setting - social studies, mathematics and music, in addition to computer education.

With a framework in mind, computer education teachers met with teachers of eighth grade social studies to formulate a cooperative project in which a series of questions provides the foundation for student projects designed to synthesize word processing skills, research/library skills, and problem solving /programming skills using *LogoWriter*.

From those meetings came the development of project outlines as seen from the perspectives of each discipline, preliminary sample questions, and the promise of staff training, especially in the area of telecommunications.

STUDENT PREPARATION

Students prepare for the *LogoWriter*/ Telecommunications/ Social Studies project by completing several projects that exhibit their skills in designing graphics, programming the word processing and labelling modes of *LogoWriter*, using multiple turtles, and combining of pages for an extended "video" project.

Preparation projects provide a foundation in the use of variables and the Total Turtle Trip Theorem - understandings essential for concise and efficient procedural programming. Students are also involved in the making of Logo quilts as a group project in which four students design and execute individual "quilt" squares within the context of a larger quilt project. Students must collaborate in the planning of the overall design or set of designs, the naming of individual procedures and within the superprocedure, the "sewing" of the quilt squares into a composite quilt, and in the problem solving encountered within the designs of individual squares. (Figures 1 & 2)

Further preparatory projects involve the use of animated graphics and illustrated text (Figure 3), programmed placement of text on the screen in the form of a letterhead and business memo (Figures 4 & 5), and adventure stories, all written as top-down planned projects.

Students acquaint themselves with the terminology and concepts associated with telecommunications by working with *The Information Connection*, including how to log on and log off, and to develop online research skills.

PROGRAM DESCRIPTION

Students are first given an outline of the project by their social studies teachers.

A series of five position questions (Figure 6) is given to students four times a year. Each student must choose one of the five questions for his/her project. With the aid of the computer education teacher, the student delineates appropriate search criteria before going online, conducts research on the specific question using the on-line resources of *Compuserve* and then writes a position statement, with bibliographic references, from the data obtained on-line.

Particular attention is given to defining search criteria - key words, including alternate, and related information - prior to student research on-line.

Staff members formulate and research the questions prior to assignment to ensure that students will be successful with their own search criteria.

Students in social studies classes who are not in computer education classes concurrently may conduct their research in a second lab located in the library/media center with the assistance of the staff of the library/media center. A second modem permits student access to the same information gained in the computer education classroom.

Students apply the factual knowledge to form a position statement, then use *LogoWriter* to create a graphics and text project based on the that same information. Students working on the same question are encouraged to work in cooperative groups of two to four persons to design the graphics project.

After choosing one of the five questions, a group of students brainstorms ideas pertaining to the question. Identifying the components of the project, making a plan and dividing responsibilities follow. Students "rough out" the graphics project on paper, then create first drafts of a superprocedure and subprocedures. Individuals type their procedures into the computer, editing and enhancing as needed. Students confer regarding the overall scope of the project and regarding procedure names. Continued communication involves problem-solving approaches especially as they link project components. Students continue to debug, polish, and refine procedures, and then to share the completed projects with class members.(Figures 7 & 8)

Time required for the project varies from five to 10 days of class time, depending on the complexity of the project and the skills of the participants.

The research and writing portions of the project typically occur before students begin the graphics and text project in Logo, requiring from two to four days. Some students use *AppleWorks* while others use *LogoWriter* as the software of choice for the position statement. For the graphics portion of the project students typically spend almost double the amount of time as on the written component. Students may also enhance the project with the use of sound/music.

STUDENT PROJECT EVALUATION

Evaluation of the cooperative projects takes several forms - including evaluation of student projects as well as evaluation of the project itself.

The position statements are evaluated by the social studies teachers for content, clarity, and persuasiveness of their writing - in addition to grammar.

The *LogoWriter* projects are evaluated by the computer education teachers for originality and for efficiency and skill in programmed procedures within the context of top-down planning.

Students are given time and opportunity to improve the efficiency of their top-down planning.

PROJECT EVALUATION

Ongoing evaluation of methods and of classroom management is a cooperative effort between the social studies and computer education teachers. We continuously evaluate the project to ascertain what we might do differently or better to accomplish our stated goals.

With the confidence gained through the use of *The Information Connection* and observing demonstrations of an online information service, *Compuserve* within our database unit, students gain an understanding of the vastness of available information, the need to be able to bring to manageable proportions that data by means of carefully reasoned search criteria, and to appreciate the greatly enhanced ability to access information.

When telecommunications - with its attendant and fundamental power to manipulate large bodies of information - is combined with a Logo environment, students gain an empowerment, a sense of control over their environments. Logo provides a graphic and dynamic process-oriented environment in which students see cause and effect, reasons for order and rules, and the nature and methods of problem-solving. While the research is inconclusive as to the measure of synthesis and cognitive transfer that occurs from Logo to other learning environments, our students gain a fluency with Logo that, at its best, frees them from the mechanical aspects of the computer to use it as the tool as personal and comfortable as a pencil.

Perhaps the greatest success of the project is found in the integral use of the computer by students in this and subsequent school assignments. Student use of the computer transcends word processing, telecommunications, or data base management. Students exhibit a synthesis of skills appropriate to individual learning styles.

Computer teachers observe considerable growth in the programming skills and understanding of problem solving concepts as our students question, "How can we make this work?" Particularly successful is the student learning that takes place on a "need-to-know " basis. Students work with and internalize new skills at a very rapid rate.

Social studies teachers view student enthusiasm as central to better writing, particularly as it pertains to the persuasive style in the position statements as well as a means for a more personal understanding of concepts and events.

Success may also be measured by the degree of communicative interaction and the enthusiasm of our students. Students are eager to share project results, including their writing, with other students in both the classroom setting and out of class. Students are frequently present in the computer room before the school day begins and after school to put finishing touches on their projects.

Success does not come without difficulties, the greatest of which is time. The project is a considerable learning experience, but we originally underestimated the time needed to complete all aspects.

We find it necessary to carefully delineate the position questions, both for clear student understanding and for the relative ease of research.

Coordination of student research for students not in computer education classes at the time of the assignment proves to be time-consuming for computer education teachers and for library personnel.

The complexity of some *LogoWriter* projects puts new demands on the computer education teachers in determining appropriate evaluation of Logo learning.

With the early versions of *LogoWriter*, some students used all available programming space. At this writing we use and enjoy the greater memory and directories of the ProDOS version of *LogoWriter*. This is especially helpful when students wish to define more than one SHAPES page.

The project proves that good keyboarding skills are necessary in a telecommunications environment. The adage "Time is money," takes on new meaning on-line!

CONCLUSIONS

We will continue to refine our expectations for the cooperative project between social studies and computer education. As the project develops, we intend to involve the language arts department in the project so that writing skills, especially in the realm of writing a position statement, may be improved.

Additionally, we hope to expand the possibilities for student projects in foreign language and science classes using telecommunications, word processing and *LogoWriter* programming skills.

REFERENCES

Abelson, Harold. *Logo for the Apple II*, Byte Books/McGraw Hill, 1982.

Abelson, Harold and diSessa, Andrea. *Turtle Geometry: the Computer as a Medium for Exploring Mathematics*, MIT Press, Cambridge, Massachusetts, 1981.

Birch, Allison. *The Logo Project Book*, Terrapin, Inc., Cambridge Massachusetts, 1986.

Bowen, Charles and Peyton, David. *How to Get the Most Out of Compuserve*, Second Edition, Bantam Books, New York, 1986.

Compuserve . Compuserve, Inc. Columbus, Ohio, 1987.

Davidson, James West and Batchelor, John E. *A History of the Republic*, Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

Electronic Mailbag: An Electric Mail Simulator, Exysm, Albuquerque, New Mexico, 1986.

Harvey, Brian. *Computer Science Logo Style*, Volume I, MIT Press, Cambridge, Massachusetts, 1985.

Harvey, Brian. *Computer Science Logo Style*, Volume II, Cambridge, Massachusetts, MIT Press, 1986.

Information Connection. Grolier Electronic Publishing, Inc., Danbury, Connecticut, 1985.

LogoWriter. Logo Computer Systems, Inc., Montreal, Quebec, 1987.

McDougall, Ann and Adams, Tony and Adams, Pauline. *Learning Logo on the Apple II*. Prentice-Hall of Australia, Sydney, 1982.

Papert, Seymour. *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books, New York, 1980.

Ross, Peter. *Introducing LOGO*, Addison-Wesley, Reading, Massachusetts, 1983.

Watt, Daniel and Watt, Molly. *Assessment of Critical Aspects of Logo Learning*, Draft Guide Book in Development, Education Development Center, Newton, Massachusetts, 1987.

Watt, Daniel. *Learning With Logo*, Byte Books/McGraw Hill, New York, 1983.

Watt, Molly and Watt, Daniel. "Logo Quilt: A Collaborative Project", *National Logo Exchange*, May 1986.

Watt, Molly and Watt, Daniel. *Teaching With Logo: Building Blocks for Learning, Addison-Wesley*, New York, 1986.

Figure 1: Quilt.Triangles

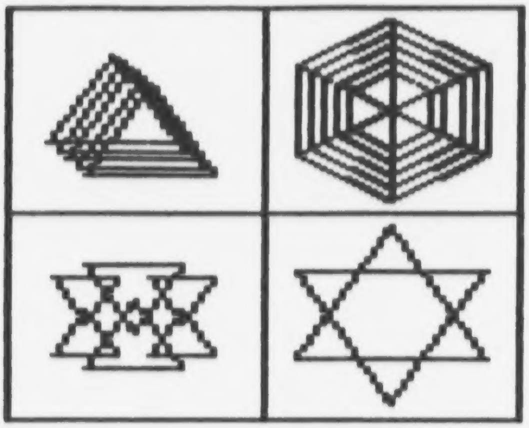


Figure 2: Music.Quilt

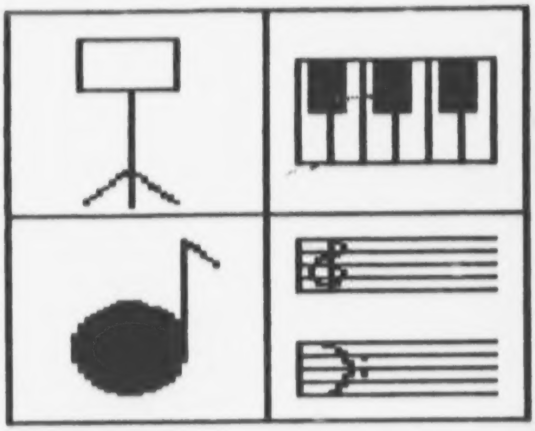


Figure 3: Animated City

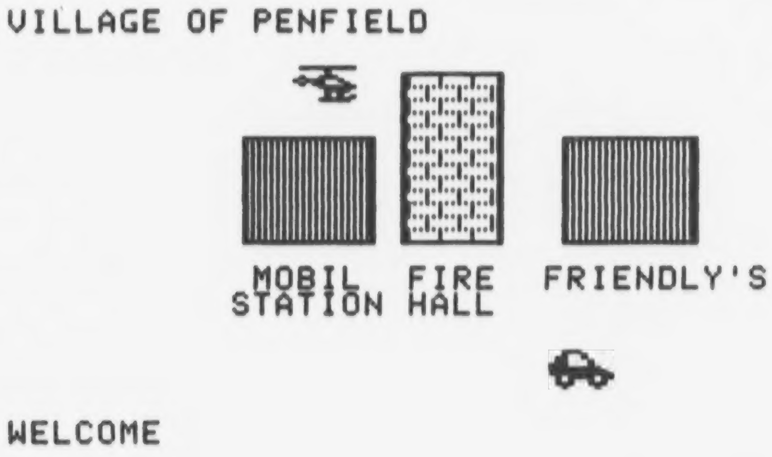
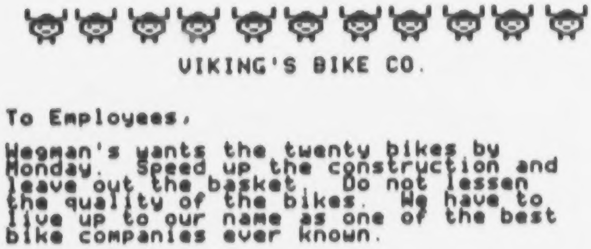


Figure 4: Letterhead.1

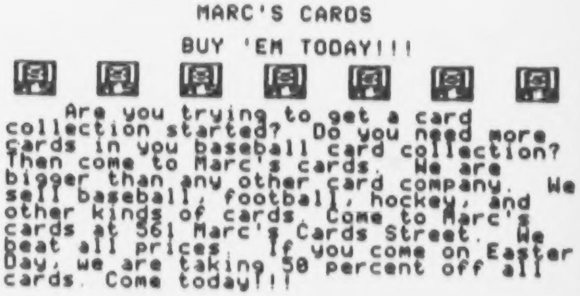


VIKING'S BIKE CO.

To Employees,

Wegman's wants the twenty bikes by Monday. Speed up the construction and leave out the basket. Do not lessen the quality of the bikes. We have to live up to our name as one of the best bike companies ever known.

Figure 5: Letterhead.2



MARC'S CARDS

BUY 'EM TODAY!!!

Are you trying to get a card collection started? Do you need more cards in you baseball card collection? Then come to Marc's cards. We are bigger than any other card company. We sell baseball, football, hockey, and other kings of cards. Come to Marc's cards at 561 Marc's Cards Street. We beat all prices. If you come on Easter Day, we are taking 50 percent off all cards. Come today!!!

FIGURE 6: POSITION QUESTIONS

1. Did the use of the atom bomb hasten the end of World War II?
2. Did the stock market crash of 1929 cause the Great Depression?
3. Was Sputnik important to education in the United States?
4. Did Lee Harvey Oswald kill President John F. Kennedy?
5. Is the United States really a "melting pot", as many people call it?
6. Looking at problems such as Love Canal, can the government really control pollution?
7. Is the space shuttle worth the cost?

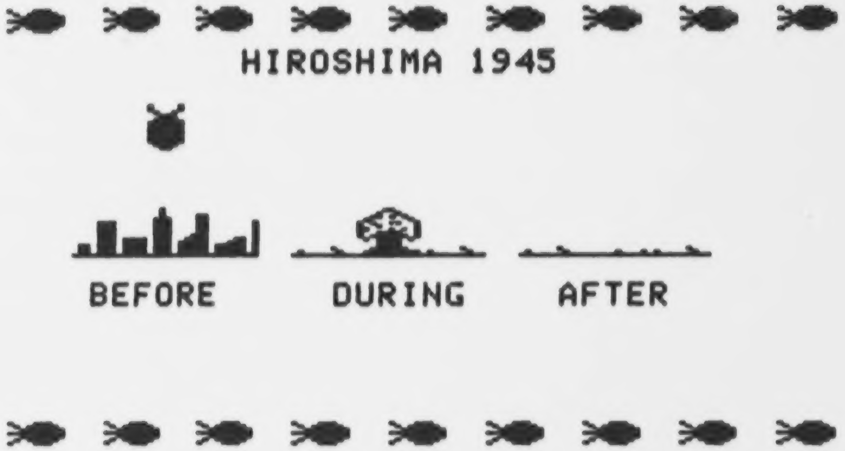
Figure 7: The Great Depression/ Dust Bowl

The Dust Bowl

The Dust Bowl was mainly in the Oklahoma and Texas areas and the adjacent parts of Colorado, New Mexico, and Kansas. In those areas were light soil, low rainfall, and high winds. The first white settlers often had livestock grazing the land. In the 20th century farmers planted winter wheat.

Between 1934-1937 there was a great drought. Much soil was picked up by the winds. Roads and houses were buried. More than half of the population left the area. The federal government replanted grass, trees, and introduced scientific agricultural methods. farming was soon possible again. the Dust Bowl areas had other less severe droughts in the 1950's and 1960's.

Figure 8: Hiroshima



In August 1945, Japan seemed to be winning the war. Many people on both sides were being killed and the outcome of the war was anything but certain.

As Vice - president, Harry Truman had not been informed about the experiments in nuclear weapons being done at Los Alamos, New Mexico. When President Roosevelt died, Truman learned about the bombs.

On August 6, 1945, Truman ordered the first atomic bomb to be dropped on Hiroshima, Japan. Two days later a second bomb was dropped on Nagasaki, Japan.

The bombs nearly wiped out the entire cities of Hiroshima and Nagasaki.

Faced with almost total destruction, the Japanese opened peace talks and surrendered on August 14

PRE-COLLEGE POTPOURRI

David N. McClintock
James Morgan
Indu Varma

ABSTRACT: Databases – Let's Get Relational!
David N. McClintock, Specialist, Instructional Support, The Department of Public Instruction, State of DE

Generally speaking, database programs come in two "flavors" – flat-file and relational . In flat-file databases, information is contained in records which are stored in one file. A good example of a flat-file database is a simple rolodex of cards, with each card containing a name, address, and phone number of an important colleague or business associate.

The information in a relational database is contained in records that are stored in many "related" files. A good way to understand this concept is to look at the structure of the database I created to track Delaware teacher inservice course registrations.

The Delaware Inservice Registration Database consists of a file containing data about each school district in Delaware, a file for each school building, and a file for each teacher. Each school building "belongs to" a school district. Each teacher "belongs to" a school building, and also "belongs to" (is related to) a district – hence, the term relational database.

From this, one may get a clear idea of what a relational database is. But why is it better to use a relational database in this instance than to use a flat-file database – especially since it is generally so much easier to learn and use flat-file databases? The answer, in a nutshell, is efficiency.

To use a flat-file database to register one teacher for six different courses, one would have to enter the name, address, district, and building information six times. By using a relational database to register the same teacher for six different courses, this information would only have to be entered once!

Indeed, the Delaware Inservice Registration Database is so efficient that one secretary handles the "whole shooting match" for all inservice course registrations state-wide. During any single inservice cycle, that amounts to about 1400 registrations from 700 teachers for 85 inservice courses.

Also, more and better services are provided. At the press of a mouse button, class rosters are printed, reminder notices are printed and sent to registrants, notices for people placed on waiting lists are printed and sent, notices for people enrolled in cancelled classes are printed and sent, and reports for the Delaware State Board of Education are printed in a fraction of the time it took to compile them before (by hand).

Relational databases can be used to make many administrative tasks more efficient. For instance, an information system with files for discipline, grades, attendance, etc. related to a student file is an ideal application to be handled by a relational database. A staff information system and an accounting system are also target-areas for relational databases. The possibilities are almost endless!

ABSTRACT: Teacher and Student Attitudes Toward Computer Access

James M. Morgan, PhD, Cincinnati Public Schools, Education Center, Cincinnati, OH 45202

The problem of fair access to computers for instruction is a continuing concern for educators. Past studies have found inequities in access to classroom computers based on race, sex, and socio-economic factors (Becker, 1986). This study was done to determine if differences in availability of computers among schools are related to teacher and student attitudes toward use of computers for instruction. Teachers and students in a medium-sized Midwestern urban school district were surveyed in spring of 1987 regarding their attitudes toward the use of computers for instruction. The survey results were correlated with the number of computers available in the district's schools and corrected for the varying sizes of the schools.

The Cincinnati Public Schools annually surveys its teaching staff and student body on a number of issues current in the schools such as achievement, discipline, and so forth. Included on the surveys are items about the instructional use of computers, including questions related to accessibility, degree of use, and effectiveness of computers. All 2,500 teachers in the school district are surveyed while the 52,000 students are sampled at Grades 6, 9, and 11. The district also annually updates its inventory of computers used for instruction in each school in the spring of the year.

Results of the survey for the computer-related

items showed that teachers in larger schools generally felt there was less use of computers for instruction in their schools and that computers were less effective in instruction. This finding was true for larger schools at all levels--elementary and secondary. However, when the survey results were correlated with the actual number of computers in the schools (corrected for the size of the school), all teachers were found to be generally positive toward the use of computers for instruction and toward the effectiveness of their use. Correlations for secondary teachers were in the .32-.34 range, while correlations for elementary teachers were in the .52-.57 range.

Results of the student survey paralleled those of the teachers. Students in larger schools reported less access to computers, were less interested in using computers, and were less likely to believe that students "learn a lot from computers" than students in smaller schools. However, when the students' survey results were correlated with the actual number of computers in their schools (again corrected for size of school), correlations were in the range of .48-.52.

These results suggest that teacher and student attitudes toward computers are related to the number of computers available to them in their schools--teachers and students in schools with lower student to computer ratios were generally more positive toward computers. The correlations are also being examined by race and sex to see if differences in attitudes are influenced by these variables. A full set of the statistical results will be available in the paper presented.

ABSTRACT: Computers and Curriculum Districtwide Involvement of Teachers

Indu Varma, Marshview Middle School, Sackville, NB, Canada

Integrating computer usage in the existing school curricula is the challenge faced by the educators today. The utility of computers as a powerful educational tool has been proven beyond doubt. The problem that we are faced with is how to effectively incorporate the use of this versatile tool in our teaching.

Districtwide involvement of teachers is highly desirable to ensure uniformity of educational experiences. It assists in the proper coordination of instruction at the various levels. Since computers have applications in virtually every subject area, it is best to involve classroom teachers in enhancing the existing curricula instead of using computer specialists.

However, there are certain important factors to be considered:

1. Time constraints on teachers;
2. Initial fear or reluctance to venture into a new field; and
3. Lack of computer experience or knowledge,

In New Brunswick's School District 14, our experience has been very positive in getting the teachers involved. The integration of computer usage in teaching began at the third grade level three years ago. We have, by now, completed implementation up to Grade 5 level. The following elements have contributed to our success:

1. Two teachers, who are in touch with the reality of a classroom, have written the curriculum;
2. Input at the grass roots level has made the teachers feel that they are a part of the whole project;
3. Activities with step-by-step instructions provided in the curriculum guide have made it easier for teachers to use computers in their teaching;
4. The hardware and software is made available to teach the program;
5. Hands-on workshops are provided for the teachers periodically; and
6. A key person is there at each school to assist teachers with the program.

A five-step model, shown below, has been used in the planning and implementation of our program.

STEPS	STAGE 1	STAGE 2	STAGE 3
1. Needs Assessment	Set up a committee	Needs assessment	Establish priorities
2. Gather Resources	Choose a grade level to start	Get input from teachers involved	Select writers
3. Write Curriculum	Gather materials and equipment	Write sample lesson plans	Prepare the Curriculum
4. Set up the Labs	Seek finances for the project	Acquire suitable hardware, software	One key person in each school
5. Train Teachers	Workshops for teachers	Feedback from the teachers	Further in-service

EDUCATIONAL TECHNOLOGY SYSTEMS/GRADUATE EDUCATION PROGRAMS

Reg Hinely, Gerald Ponder, Bill Atkins
Peter H. Martorella
Gary W. Tubb, Frank Breit
Joyce Hart, David Burrows

ABSTRACT: School-University Collaboration: A Field-based Doctorate in Curriculum and Instruction with Emphasis on Computer Information Systems

Reg Hinely and Gerald Ponder, College of Education, The University of North Texas, Denton, TX; and Bill Atkins, Plano Independent School District, Plano, TX

A major manpower need of school districts is for persons who combine substantive knowledge in curriculum and instruction with comprehensive knowledge and understanding of emerging developments in educational technology. Too often important policy decisions about the purchase and use of new technology are made on the advice of computer experts who have little knowledge of how schools work, or by an expert in curriculum and instruction with little knowledge of hardware and software options and alternatives. On the initiative of, and in collaboration with the Plano Independent School District, the College of Education at the University of North Texas has developed an interdisplinary, field-based doctorate specifically designed to meet this need.

The program of studies stresses the knowledge base of the fields of curriculum and instruction with emphasis on applications in educational information systems. The program is a joint effort of the departments of Computer Education and Cognitive Systems, Educational Administration, Research, Elementary, and Secondary Education. In addition to the university and the public school district, the program draws on the resources of the technological, corporate, and other educational entities in the North Texas area. Since the knowledge base for the program is exceptionally fluid and is expanding at a rapid rate, the program features a sound core in curriculum and instruction, in computer education, significant work in seminar formats using readings, research, site visitation, and resource persons with expertise in relevant technology systems.

Twenty students are currently enrolled in the program and have completed the first 12 hours of course work. Numerous requests from individuals and from school districts have been received requesting information and/or seeking admission to the program. The core program consists of a cohort of students moving through the various program components together with additional students allowed to take or to audit selected courses. Program evaluation will be used to modify the pilot program as well as to restructure and refine existing doctoral programs.

ABSTRACT: Developing a New Generation of Computer Specialists for Schools

Peter H. Martorella, Department of Curriculum and Instruction, North Carolina State University, Raleigh, NC

Recently the Department of Curriculum and Instruction at North Carolina State University developed, obtained state certification for, and implemented a new graduate degree program track to prepare a new category of professionals "Instructional Technology Specialists - Computers." The program draws cooperatively upon the resources of both the Department of Curriculum and Instruction and the Department of Computer Science.

The role of the ITS-C may vary, ranging from that of an educational leader who coordinates the overall computer technology program within a school system or an individual school, to one who concentrates on a certain grade level, a certain subject area, or several areas of the curriculum. Such an instructional specialist may be identified by a variety of titles: for example, computer coordinator, instructional computer specialist, computer resource teacher, instructional specialist for computers, or education supervisor. This instructional specialist is distinct from individuals who are preparing exclusively to teach computer science courses in secondary schools.

The ITS-C individual would be expected to have job responsibilities within a school setting that include skills related to computer operations, curriculum development and implementation, hardware and courseware evaluation, aiding teachers and other educators with applications of computers to instruction, and supervision of related personnel facilities.

The specified educational objectives of the new program track are grouped under the five broad categories listed below:

1. Knowledge and Understanding of Computer Technology and its Application to the K-12

Curriculum
2. Program Planning and Implementation Skills
3. Evaluation, Selection, and Integration of Hardware and Courseware into the Ongoing Instructional Program
4. Production and Utilization of Computer-Related Resources
5. Organization and Management of Personnel and Facilities

ABSTRACT: Instructional Computing: A Graduate Education Curriculum - 5 year evaluation

Gary W. Tubb, Ph.D. and Frank Breit, Ph.D., Instructional Computing Department, University of South Florida, Tampa, FL

This project presentation addresses the objectives and evaluation of a graduate program in Instructional Computing in Florida. The objectives of the graduate program as well as of each individual course will be shared with participants of NECC. The results of an indepth evaluation of alumni, current students and faculty will be presented.

The masters program of courses in Instructional Computing (21 semsester hours of educational computing courses) at the University of South Florida was formally approved in 1983. Over the last 5 years, the number of active graduate students has ranged from 25 to 125 per year. Many of these students have become teachers of computing in public schools or in industry, county supervisors of Instructional Computing, community college teachers and administrators, or professors of education. In some cases, the curriculum did not meet the immediate needs of students enrolled in the program. The short-comings of the curriculum focused upon the relevance of educational foundation courses to the preceived needs of the computing teacher. Specific details of working jointly with the fourndations faculty to develop relevant technology based content courses will be shared with the NECC participants.

Teacher certification of K-12 Instructional Computing teachers in Florida emerged from the original courses developed in the University of South Florida's Masters program.

ABSTRACT: Curriculum/Instructional System

Joyce L. Hart, Curriculum Services Officer, and Dr. David Burrows, Information Systems Officer, Florida Community College at Jacksonville, Jacksonville, FL

Florida Community College at Jacksonville has an administration building and four campuses geographically spread out over Duval County, to serve both Duval and Nassau counties on Florida's Northeast (Atlantic) Coast. In addition, the college has established hundreds of off-campus centers of instruction.

The college emphasizes a one-college goal to help accompish this, among others, is the computerization of college data for management, application, processing, review, and college-wide communication. The Automated/Integrated Curriculum/Instructional-based System, designed by Joyce Hart, addresses that goal.

The system is a model in communication, curriculum innovation and development, and instructional resources. The Curriculum/Instructional System project to automate and integrate three major systems (the Curriculum Action Process, Official Course Outlines, and the College Catalog) was approved by the college, and fully developed and implemented during the 1987-88 academic year.

The college already had established a College-wide Curriculum Action Process that supported the one-college concept and structure. It required manually selecting and filling out forms to accomplish a desired result, and several manual modifications as a result of time-consuming college-wide review of the duplicated hard copies. The elements that comprise the contents of the Course Outlines and the College Catalog are all developed, deleted, or updated through the Curriculum Action Process. The manual process required copying, noting, storing, and rekeying appropriate elements from the proposals to eventually get them into the appropriate files/systems.

Conversion from the manual to the automated system took several months and included:

1. Conversion of word processed Outlines, Curriculum Process forms, and College Catalog content to RFT format and uploading to the mainframe computer;

2. Technical designation of mainframe storage, set-up, and access;

3. Internal testing;

4. College-wide demonstration and testing of each of three major portions of the project;

5. Trouble-shooting systematic and technical problems;

6. Writing user and manager instructions;

7. Setting up and testing security; and
8. Setting up adequate user access.

With the implementation of this project, the Curriculum Action Process, the Official Course Outlines, and the College Catalog are accessible for display by electronic mail college-wide via terminal or microcomputer connected to the mainframe. The Curriculum Process shells (forms) are accessed, the appropriate combination chosen, filled in on-line, and compiled into a proposal from any mainframe connection, electronically transmitted college-wide for review, and subsequently updated by the originator. Finally, specific elements from the approved curriculum proposals are copied from the file into the Course Outlines and the College Catalog files without rekeying the details.

PRODUCING AND USING SPEECH-BASED SOFTWARE: A PANEL PRESENTATION

Lynne Anderson-Inman
Associate Professor, College of Education
University of Oregon
Eugene, OR

ABSTRACT

One of the most exciting advances in computer-based technology for education is the increasing availability of quality software with speech output. This presentation provides an overview of current information and issues related to the production and use of speech-based software. Panelists bring a wide range of experiences producing and/or using software with speech output and will share diverse views concerning appropriate applications and critical issues related to this emerging technology.

Bill Adler from Street Electronics will talk about the technical aspects of producing software with speech output. The relative advantages and disadvantages of three approaches to the production of computer-based speech will be compared: synthesized speech, digitized speech and linear predictive coding (LPC). In addition, he will suggest future possibilities for technical improvements in the production of speech for educational applications, with attention to the use of speech in CD ROM.

Mary Cron from First Byte and Michael HIllinger from Houghton Mifflin will provide a publisher's perspective on the development of speech-based software. After an overview of the types of programs available with speech output they will discuss the various roles for the use of speech in educational software and issues related to production and design. Topics addressed will include the effects of speech on software design and the user interface, as well as the effects of computer-based speech on production costs and product

Dick Olson will share research conducted on the intelligibility of synthesized speech, addressing the importance of high quality speech for specific educational applications and the range of intelligibility produced by different systems. Dr. Olson will also share research on the effects of using software with speech feedback to improve the reading performance of disabled readers. Future needs with respect to research on speech-based software and reading instruction will be suggested.

Bonnie Prohask and Maria Gonzalez-Baker will explore various educational applications for software with speech output. Some of the applications to be discussed are the role of speech-based software in: (a) reading and writing instruction, (b) teaching young, at-risk students, (c) stimulating second language development, and (d) teaching content area knowledge to bilingual children. Issues and problems related to classroom implementation of software with speech output will also be addressed; e.g., the effects of speech-based software on the teacher's role, students' reactions to computer-based speech, and suggestions for improving the quality of software containing speech output.

Lynne Anderson-Inman will conclude the panel presentations by sharing a list of factors to be considered when evaluating speech-based software. Two types of factors are suggested as important: (a) factors related to the purpose and use of the software being evaluated and (b) factors related to the software's construction, specifically the perceived role for speech output in a given program. To assist conference participants in examining software with speech output, information will be provided concerning software developers with speech-based products.

PANELISTS

Bill Adler
Street Electronics
Santa Barbara, CA

Mary Cron
First Byte
Long Beach, CA

Michael Hillinger
Houghton Mifflin Co., Educational Software Division
Hanover, NH

Richard Olson
Colorado State University
Boulder, CO

Bonnie Prohask
Franklin Elementary School
Madison, WI

Varia Gonzalez-Baker
P Falcon International (PFI)
San Antonio, TX

TOOLS FOR TEACHERS AND TELECOMMUNICATIONS

Owen F. Gaede
Michael Land
Roberta Sloan Little

ABSTRACT: Development of a State-Wide CBE Network in Maine Using NovaNET(sm)

Owen F. Gaede, Ph.D., Director, Office of University Innovations, University of Maine, Alumni Hall, Orono, ME

In the fall of 1986, Dr. Don Bitzer of the Computer-based Education Research Laboratory (CERL) at the University of Illinois announced the development of NovaNET(sm), an experimental satellite-based national CBE network. NovaNET(sm) is being designed to accommodate thousands of simultaneous users and will resemble closely the PLATO system. Dr. Bitzer is the "father" of PLATO, and has been involved in CBE development for over 25 years.

The University of Maine is attempting to develop a state-wide CBE network using NovaNET(sm) to link public schools, vocational-technical institutes, universities, colleges, and the general public. The NoveNET(sm) satellite downlink became operational during November, 1987 and the Maine network may reach several hundred simultaneous users by the NECC conference in June, 1988.

This project report will summarize progress in the implementation of this state-wide CBE network as of the end of the 1987-88 academic year. The report will include a summary of the costs incurred since the inception of the project, usage to date, and projected applications over the 1988-89 academic year. Advantages and disadvantages of NovaNET(sm) over other alternatives, such as stand-alone microcomputers and PLATO, will be discussed.

ABSTRACT: Using a Classroom Database Program to Prepare Teacher-Generated Tests

Michael Land, Department of Computer Education, National College of Education, Evanston, IL

Innovations may come and go but tests, quizzes, and worksheets are a continuing part of most classrooms. Any inexpensive and easy-to-use system or technique to aid in the preparation of quizzes is welcomed by most teachers.

In this session, I will present some of the tasks of a teacher training sequence that I use at National College of Education using a database program to prepare quizzes, tests, and worksheets.

The tasks in the training sequence include:

- Designing quiz bank templates.
- Choosing and entering items into the quiz or test bank.
- Searching (identify) items in the database appropriate for a particular quiz.
- Preparing an on-screen preview of items selected.
- Preparing variations of the same test, either by rearranging the order of questions or by rearranging multiple-choice responses or both.
- Preparing an answer from the database.
- Placing the test items into a wordprocessing file to prepare the final version of the test.

We use *AppleWorks* in our training but any database program can be used. However, each database program has its own peculiar limitations and strengths. Contact the presenter if you would like a hardcopy or a disk copy of the template files.

ABSTRACT: Tools for Teachers

Roberta Sloan Little, Curriculum Software Consultant, UNC-Educational Computing Service, Research Triangle Park, NC

Teachers of every discipline and at all levels must perform many of the same routine tasks. They prepare course schedules, class notes, and quizzes, maintain student records, and keep track of appointments, deadlines, holidays and breaks. Software is available which will assist teachers with these and other mundane tasks. With budget concerns in mind, UNC-ECS has grouped together a selection of MS/PC DOS shareware and low-cost software and labeled the collection "Tools for Teachers." Included in the collection are software packages which meet the following specific needs:

Outline processor - produce course outlines, class notes, book lists and to-do lists

Spreadsheet program - maintain class records and design seating charts

Gradebook programs - maintain class records

Calendar program - track dates and schedules of events

Quiz generators - administer online quizzes or create printed quizzes

Editor - create test bank of questions for quiz programs

Graphics program - produce graphics and charts

Presentation software - create and present on-screen slide shows

Word processor - grade machine-readable text

The presentation includes a computer slide show of samples from the software in the "Tools for Teachers" collection, a demonstration of how the slide show itself was created, and a discussion of the various software packages.

COMPUTING IN THE U.S.S.R.

Karen Billings, *Chair*
Vice-chair, ACM/SIGCUE
Logo Computer Systems Inc.
Boston, MA

ABSTRACT

The United States Information Agency is currently sponsoring a traveling technology exposition in the United Soviet Socialist Republics which includes a section on computers in education. Several people with experience in the field of computers in education have been invited to participate as "Specialists-in-residence." Each specialist spends five to six weeks in a Soviet city working with the exposition.

The "Specialist-in-residence" works in the exhibit computer room meeting with people and small groups to answer questions and give talks. They also have the opportunity to visit Soviet schools, computer clubs, and universities.

The panel consists of several people who have been on their tour. Cities visited by members of the panel include Moscow, Rostov, Tashkent and Kiev. Discussion will be on the state of Soviet educational computing as they had the opportunity to observe and what the representatives of the United States presented in the exposition.

PANELISTS

Frank Connally
The American University
Washington, DC

Tom Lough
University of Virginia
Charlottesville, VA

Cheryl Weiner
Spectrum Interactive, Inc.
Burlington, MA

SPONSOR: SIGCUE

THE REQUIRED COMPUTER EDUCATION COURSE: WHERE ARE WE AND WHERE ARE WE HEADED?

Pamela Moore, *Chair*
Eastern Michigan University
Ypsilanti, MI

ABSTRACT

In June 1987, at NECC '87, the Special Interest Group on Teacher Education of the ICCE established the Required Computing Course Committee. Its mission had two major thrusts: 1) to survey a national sampling of teacher education institutions to discover what was included in the computer education course required of all teacher candidates; and 2) to predict national trends in the directions the course was taking. Most of the committee met a second time at NECC '87 and decided on a general plan of attack. We would a) develop a questionnaire to get as much information as possible in the most concise manner, b) send the questionnaire to all current SIGTE members, c) collate results, and d) disseminate findings.

The discussion will cover the following points:

1. Development of the questionnaire
2. Methodology for conducting and evaluating the survey
3. Results of the survey regarding
    a) What is covered in the basic computer education course for teachers?
    b) National and regional trends
    c) Future implications on computer education for teachers

PANELISTS

Joyce S. Friske
Oklahoma State University
Stillwater, OK

Mildred D. Lintner
Eastern Michigan University
Ypsilanti, MI

Bonnie Mathies
Wright State University
Dayton, OH

Lajeane Thomas
Louisiana Technology University
Ruston, LA

James Wiebe
California State University
Los Angeles, CA

Charles Mlynarczyk
State University College
Potsdam, NY

SPONSOR: ICCE

ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING; MODEL COURSE OUTLINE

Robert Taylor, *Chair*
Project Director
Teachers College, Columbia University
New York, NY

ABSTRACT

Teachers just now emerging from preservice education programs are often little more prepared than the experienced teachers they will join to use computers in their classroom. They are emerging as an addition to the problem rather than part of the solution. While some colleges are addressing technology training well, for the most part they are floundering, unsure of what to teach or how to gain the resources to teach it.

Based upon the fourteen background papers recently published in the SIGCUE Outlook and the intense critique of participants at a working conference held at Teachers College, a model preservice course in educational computing was developed.

The establishment of this interim preservice computer course is a first step in insuring that schools of education generally can provide a program so all emerging educators have the needed competencies to be effective and innovative users and consumers of technology. It is our hope that these newly trained teachers will, through their successful use of computers, become models for and aids in the retraining of existing teachers, rather than an addition to the body of teachers that already need to be retrained.

PANELISTS

Howard Budin
Teachers College, Columbia Univ.
New York, NY

Sheila Cory
Chapel Hill - Carrboro City School District
Chapel Hill, NC

John Lawson
Towson State University
Baltimore, MD

David Moursund
International Council for Computers in Education
Eugene, OR

SPONSOR: SIGCUE

HYPERTEXT STUDY GUIDES: AN AID TO COMPREHENSION

Randy Boone & Kyle Higgins
Experimental Education Unit
University of Washington
Seattle, WA

ABSTRACT

The purpose of this project, *Hypertext Study Guides: An Aid to Comprehension*, was to develop supplementary, highly interactive, computer based reading materials designed to facilitate the successful participation of handicapped learners in a regular high school social studies classroom.

The project sought to identify methods for increasing the amount of direct content reading instruction for low performing and handicapped students within the regular education setting. Important concerns within this setting were: (a) increasing instructional time without increasing the demands on the teacher's already limited time, and (b) creating a system which could offer access to new curriculum or enhance an already established curriculum. Computer assisted instruction was chosen as the means for integrating this additional instructional time.

A *hypertext* format, written in the *Guide* authoring system for the Macintosh computer, was chosen for the CAI lessons. In a hypertext CAI lesson, words, parts of words, graphics, or parts of graphics can include more than what first appears on the surface layer of the screen. A hypertext page might be thought of as a composite of several sheets of transparency film overlaying each other, the top layer not only providing the original text to be read but also serving as a menu for accessing additional information, such as getting help with unknown words, or following tutorial strategies for building comprehension.

Positioning the cursor on specific areas of the top layer hypertext page makes the appropriate layers beneath accessible. Selecting a word with the cursor could give, for example, a choice of the word's definition, pronunciation, etymology, or provide a graphic related to the word. Further button selection could take the user to the third and subsequent layers of information. This layering of text is virtually limitless as far as the technology is concerned. Depth of information should be controlled, however, to provide for optimum access by the students.

These unique lessons provided the students with a dynamic, nonlinear reading environment offering additional information about words and concepts from the social studies text. The hypertext reading environment provided the students with access to special enhancements within a familiar context. This differs from traditional CAI which often presents lessons on isolated skills not specifically related to textbook content or the teacher's instructional style.

The use of hypertext CAI to increase comprehension competency provides targeted help for at-risk students in areas in which they may be experiencing difficulty. It provides a means of modifying traditional material within the regular classroom environment. This modification presents a means by which more students will learn to reason effectively about what they read, analyze what they read, and elaborate upon and extend the ideas in their reading material. The use of the microcomputer medium provides a means of maximizing the active participation of the student in reading, as well as a consistent curriculum-based assessment of student progress. In this manner, the student and the teacher are constantly updated as to reading progress, student participation, and student success in meeting the reading demands within the mainstreamed environment of a content area classroom.

The software which was developed and field tested during this study provided a teacher within the regular education classroom environment the means to increase instructional efficiency in the classroom as well as ensure educationally productive time for all students. This is particularly important for the handicapped learner in that it provides the added exposure, practice, and actively engaged time necessary to remain competitive within the regular educational setting.

The advent of microcomputers has fueled an explosive growth in educational computer usage. Although we are witness to this remarkable rate of technological growth in our schools, we have been unable to establish a sound research base from which to evaluate the use of CAI. Definitive data relative to the pedagogical effectiveness of CAI as an instructional medium have remained elusive.

Preliminary data analysis from this experimental study, however, indicates that for students identified as

below average, the hypertext lessons were successful as an alternative to the conventional forms of reading, lecture, and note taking, and as supplementary instruction.

A MODEL FOR THE INTEGRATION OF COMPUTERS INTO THE CLASSROOMS OF THE SEVERELY LEARNING DISABLED

Constance McAvoy
Greater Victoria School District
Victoria, BC, Canada

ABSTRACT

The focus of this presentation will be on the model of inservice and support used to integrate computers into the curriculum of the severely learning disabled child. The outcomes of the project evaluation will also be discussed.

Through a Government of British Columbia "Funds for Excellence" grant offering, the teachers involved with the Severely Learning Disabled students Grades K-7 in the Greater Victoria School District submitted a proposal to integrate computers into their curriculum.

The proposal included provision of hardware, provision of software, and provision of a coordinator to provide inservice, classroom support, and a resource manual.

The idea was to give the learning disabled child access to a tool that provided the technology to bridge the gap between his/her skills and those of the average school child. The idea was also to give the educators of these children the training and support necessary to make these machines functional tools within the special education curriculum.

The profile of the severely learning disabled child:
A child who is of average or above average intelligence but is not working up to his/her potential because of specific learning disabilities. The student is unable to achieve success in regular classes. The student exhibits a significant discrepancy between estimated learning potential and actual performace as determined by standardized score comparisons. The student exhibits problems in attention, perception, symbolization and the understanding and use of spoken or written language, which may be manifested in difficulties in thinking, listening, talking, reading, writing, spelling or computing. The student does not demonstrate behavior or emotional difficulties as the major presenting problem.

An important emphasis of the project was that students would gain proficiency in written communication skills using word processing. Students all succeeded in developing keyboard familiarity and technical competency with a word processor and in generating presentable text using the word processor. Junior and Senior elementary students used spelling checker software which did result in an improvement in spelling in the output generated with the word processor compared to pen and paper.

Students also interacted regularly with a variety of different software packages, including some mathematics and reading drills, and some which were oriented toward problem solving. Overall the project which ran from January 1987 to January 1988 provided an excellent experience for the students.

Teachers displayed considerable professional growth during the project. All developed and strengthened their technical competencies with computers and all developed strategies to effectively manage the integration of computers into their ongoing instructional planning and delivery. The teachers also gained experience in software evaluation. The response of the teachers was strong and positive and its impact on them as professionals was also very strong and positive.

UNIVERSITY POTPOURRI

John E. Cook
Paul Plourde
William Wehrs, Thomas D Amico, Dawn Gering, Irving Locketz, Valerie Schmitz
Mitchell Batoff

ABSTRACT: Graduation and Advisement System (GRADS)

John E. Cook, PhD, Dean, School of Business, SUNY College of Technology at Utica/Rome, Utica, NY

The college developed and implemented a computer assisted advising and administrative planning system. The GRADS system was designed to establish a baseline for academic planning for each entering student and supply an analysis for that student in regard to course and degree planning. It was also expected to supply administrative information with a complete inventory of individual student curriculum requirements. SUNY at Utica/Rome has unique problems with an advising system because all students are transfers. Advising was a very labor intensive process.

Systems at other colleges and commercial software were reviewed and an in-house system was developed. The system was tested on the business academic program representing about 25% of the students and has been operational for five years. GRADS has been operational on the entire campus about three years. A number of benefits have resulted and a number of concerns still exist.

A principle benefit is the sharpening of curriculum requirements and academic regulations. Under the previous labor intensive system, advisors often determined the academic program for individual students. In some departments students were treated unevenly and sometimes unfairly. Surveys showed dissatisfaction with "advising" within these departments. Survey results have improved with implementation of GRADS.

A second benefit has been the fostering of independent behavior from most students. All students are at least junior level, and most are in professional majors such as business, engineering technology, computer science, nursing, and so forth where mature behavior is consistent with curricular objectives. Students are able to do their own academic planning with GRADS. Some students who are not yet ready to assume such an independent role find shorter advising queues.

The third major benefit of the GRADS system has been useful administrative planning information.

Problems have interfered with implementation or performance of the system. One human problem is a select-faculty vested interest in student dependence. Some faculty want the opportunity to coerce students in adademic choices. One major advising complaint from some students were phantom "required" courses. Student GRADS reports stopped most of these problems. A second major problem is an inertia for change. Several procedures that the GRADS system rendered obsolete were continued simply from this inertia.

The system also suffered sabotage. One academic program submitted inappropriate data on students so resulting outputs were inaccurate and damaged system credibility. Ignorance delayed development and implementation of GRADS. A fear of automation was involved. The final problem was system or program errors. GRADS has to be grafted onto existing systems such as a student records system. Current hardware and personnel were used. Many problems were corrected but some must wait for the second generation system now being planned.

ABSTRACT: A Four-Year Retrospective on Requiring Microcomputers

Dr. Paul J. Plourde, Professor, Department of Computer Information Systems, Bentley College, Waltham, MA

In the fall of 1987, all freshmen and sophomores at Bentley College in Waltham, Massachusetts, were issued a portable microcomputer. In addition, all juniors and seniors were provided with the opportunity to rent or purchase a desktop microcomputer. As of this date, 892 micros have been issued to freshmen and 842 have been issued to sophomores. In addition, over two hundred have been sold or rented to juniors and seniors.

This year's implementation of the required micro policy was the fourth year that the college has issued computers to all or a part of the freshmen and/or sophomore class. This program was initiated in 1984-85 when a pilot project of 115 freshmen was conducted. These students were issued portable computers equipped with 256K of memory, a disk drive and a printer. They were also issued BASIC, LOTUS, MEMOMAKER and an Integrated Accounting Package. They were required to use the computer and the software in three freshman courses: Accounting, English and the Introductory Course in Computer Information Systems.

From the outset, we were concerned about the impact of the use of micros for coursework and for personal use, and we developed survey instruments that have been administered at the beginning and the end of each year. These sought to determine changes in student attitudes towards computing. Based on the initial results of these surveys and other studies designed to measure the impact of computer use in courses, the program was extended to include all freshmen in 1985-86, and further extended to include all sophomores as well as freshmen in 1986-87 and 1987-88.

The description of the pilot project and the extension of the program to the entire freshman class were reported at the National Educational Computing Conference in 1986 and appear on pages 107-111 of those proceedings.

Throughout this four-year period, we have had an opportunity to evaluate the program from several perspectives and we have made a number of changes to the:

1. Curriculum and courses that use computers;
2. Hardware and software that is issued and used;
3. Methods used to introduce the students to the computer and associated peripherals;
4. Methods used to familiarize students with the software and associated workbooks;
5. Approach used to train over 200 faculty to become computer fluent; and
6. Logistics of distributing and servicing over 2,000 microcomputers, software and documentation.

This is a report of a longitudinal study of the experience of one of the 20 or so colleges or universities that require students to acquire or use a microcomputer.

ABSTRACT: ESAU - Expert System for Advising Undergraduates

Professor William Wehrs, Thomas D Amico, Dawn Gering, Irving Locketz, and Valerie Schmitz, University of Wisconsin - LaCrosse, LaCrosse, WI 54601

Each semester students are required to see their faculty advisor prior to registration for classes in the subsequent semester. The purpose of this meeting is for the advisor to help the student make a proper choice of classes given university requirements and the preferences of the student. University requirements are documented in the university catalog. There are two significant problems with current advising procedures. First, advising expertise varies widely among the faculty. This is largely due to the dynamic nature of university programs, the length of time that the faculty member has been at the university, and the faculty member's attitude toward student advisement. Second, the actual advising session frequently involves an inefficient review of classes already completed by the student. Knowledge of previous call work is needed to determine the student's status as compared with the requirements of his or her chosen plan of study. A review is necessary if the faculty advisor does not have current records on completed classes for a given student.

ESAU is designed to supplement the current advising process and to address its two significant problems. Prior to meeting with the faculty advisor, students are required to have a consultation with ESAU. During this consultation the student provides data on preferences and completed classes as of the end of the current semester. A rule-based inference engine employs the preference and class completion information in conjunction with a set of rules (a knowledge base) largely derived from curriculum requirements specified in the catalog. The end product of the consultation session is a printed list of classes completed, and a printed suggested schedule of classes for the coming semester. These documents provide the basis for an advising dialogue between the student and his or her faculty advisor during the subsequent advising session.

A prototype of ESAU is implemented using the inexpensive expert system shell VP-EXPERT. This prototype is limited to students pursuing a concentration in Management Information Systems within the Management major at UW-LaCrosse. ESAU's significant features include a "friendly" user interface, modularity with respect to different classes of curriculum requirements (General Education, Core, Major), and ease of knowledge base maintenance as these requirements change dynamically.

ABSTRACT: Public Domain Software; Its Importance in Teacher Training; Some Subtle Points to Ponder

Dr. Mitchell E. Batoff, Professor, Jersey City State College, Jersey City, NJ

There are currently 1.2 million microcomputers in schools across the United States, with an average number per building between 15 and 23. This large investment in computers and software has created a need for inservice training of school personnel in appropriate computer use and preservice training as well.

This presentation relates to teacher training at the preservice level, or inservice training as part of a staff development effort in a school system, or as part of an inservice graduate course, or in a university as part of a faculty training effort. The common element in all of these situations is a novice who needs to be made *comfortable* and *confident*.

In any training effort one or more sessions dealing with public domain software can contribute to two of the most important facets of teacher training, viz., *comfort* and *confidence in using the equipment*. This presentation will describe how and why.

The presentation will be divided into four parts: (a) How public domain software can help the novice become comfortable and confident in using a microcomputer; some significant but subtle points to ponder, (b) Some important things to know about public domain software, (c) Two dozen sources of choice public domain software, and (d) Distribution of a sample public domain disk to all participants at this session. This will be of general value regardless of grade level or subject matter interest.

A HI-TECH LAB FOR TEACHING ENGLISH STUDENTS THE WRITING PROCESS

Frank J. Ferrari
Danville Area School District
Pine and E. Mahoning Streets
Danville, PA 17821
717-275-7574

Jean Knouse
Danville Senior High School
Northumberland Street
Danville, PA 17821
717-275-4111

ABSTRACT

A lab with twenty microcomputers networked to a laser printer was installed in a classroom at the senior high school. A modem was used to retrieve data, as well as to teleteach students incarcerated in a maximum security juvenile detention center. An interactive video system was used to provide remediation in a library resource room. Although the classroom-lab was used formally to instruct sophomore English classes, students in the high school (enrollment 552) use the lab before and after school, under the supervision of a teacher or aide. The project involves two English teachers, two computer (one Business and one Math) teachers, and one aide in the classroom lab. The librarian and aide assists in the interactive video and telecommunications. The administrative staff oversees the entire project.

Teaching students to think and to learn how to learn are primary objectives in our school district. A Hi-Tech Lab which features a networking system that produces high quality copy and telecommunications enables us to teach our students how to think and learn by doing.

OVERALL GOALS/NEEDS

The purpose of this project is to thoroughly develop our students' abilities to write well in a Hi-Tech society; to research facts through telecommunication; and to participate in desk-top publishing. These skills will be accomplished by creating a Hi-Tech English classroom to teach the writing process to all high school students. Along with teaching our high school students, the teacher will be teleteaching to a maximum security juvenile detention center. Teleteaching is an educational delivery system where an expert teacher can instruct a distant group over the existing telephone lines. Each session will also be video taped, so that students who are absent can make up the work they missed using an interactive video system. Additional lessons for remediation would also be compiled, so that a complete interactive video library would be compiled. The writing needs of our students are multifaceted. Approximately sixty-five percent of our graduates enter higher education. They need to know and be able to skillfully use the writing process. In addition to writing well, these students must be able to access data bases, through telecommunicating, to fulfill their research needs. Being able to word process and understand desk-top publishing is of maximum importance in the Hi-Tech educational environment of the future. However, these factors are of equal importance to the thirty-five percent of students who will be entering the military or the working force.

SPECIFIC PROJECT GOALS

All students will begin by working the Milliken Publishing Company software "The Writing Workshop." According to one of the many references our team discovered, an article by researcher Tom Donahue, *Times Herald*, City Editor, Michael Moore from Milliken Publishing Company in St. Louis noted, "It teaches by example." The principal of Chester Allegheny High School, Charles Chester said, "I liked the self-confidence end of it." (Donahue 1986)

Our students have access to the six to eight thousand resource documents that are housed in our high school library and a small local library. By using a modem, students will be able to access Compuserve, BRS, PennLink, and Public Domain Data Bases. This access will expand the resource documents that are available to them from eight thousand to millions. Since all students, from special ed to college prep will be required to research a topic, an impressive amount and variety of data will be retrieved.

Another goal is to teleteach the writing process to incarcerated youth. By learning to write, to word process, and to use desk-top publishing, these special students will learn valuable skills that will help them function in society. Research shows that literacy has a direct ratio to success in our social environment. Additionally, it will allow these students access to the real world without risk to the school community.

The use of interactive video for remediation is presently untapped. Using live video to exhibit the pedagogical skills of the teacher, united with the ability of the computer to develop cognitive learning, is exciting. If a student misses class or is falling behind, he can go to a resource center and, in effect, take an expert teacher with him via the interactive video system.

To integrate these learned skills, the final class project will require each student to do research using telecommunications, to complete a word processed research report or research paper, and to use desk-top publishing.

NEEDS ASSESSMENT

Our school district's administrative philosophy regarding any major program development is to include anyone who will be involved in implementing the new program in the planning process. In a complete curriculum revision, Task Force Committees have been formed for each content area; a Task Force is composed of administrators, teachers, and when possible, community leaders. The Language Arts Task Force determined, through monthly meetings, that there was a need to improve student writing, and they included the writing process in each level of the English curriculum. Other Task Force committees strongly recommended using the writing process and word processing skills to create interdepartmental papers.

For the past several years, approximately sixty-five percent of the graduating seniors have been attending institutions of higher learning. Informal surveys have indicated a large percentage of these college students have a difficult time writing the papers necessary to be successful in higher education. Many also find themselves unable to use the computerized library systems and telecommunication facilities which are now found on most university campuses.

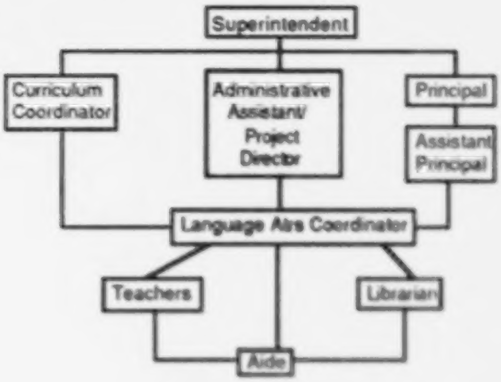
To accommodate the thirty-five percent not attending higher education, we have surveyed the business students who have graduated in the last two years. We have also surveyed local business leaders concerning their employee needs. It was not surprising to find that both groups indicated a need for graduates to have better writing skills, as well as the ability to word process information.

During the past year, we have attempted to determine the computer needs in our district in three ways: our computer initiatives coordinator has noted community and faculty suggestions; our Task Force Committees have made formal recommendations; and community, faculty, and student surveys have provided us with various kinds of helpful information. From these diverse sources, our hardware and software committees have prioritized our district needs and selected three areas of focus for the coming year. The first priority is that the program which is planned should serve as many students as possible. Secondly, as each task force creates new planned courses of study, compatible software should be purchased for computer-assisted instruction. Finally, it was determined that a Hi-Tech Lab can no longer be considered a luxury item if our students are to have the ability to compete in a Hi-Tech world.

METHOD OF CARRYING OUT THE PLAN

This program will utilize the Hi-Tech Lab/Classroom every period of every day during the school year. In addition, the Lab will be open each day before and after school. An aide who is currently working with microcomputers in our library would be commissioned to work in the Hi-Tech Lab one to three hours per day depending upon student needs. Our school district also sponsors many summer youth programs and would utilize this lab during the summer months.

The following organizational chart and explanation will detail the project.



Stage 1 - The Superintendent, Administrative Assistant/Project Director, and the Curriculum Coordinator will be responsible for the implementation of the project. They will oversee the project goals, timeline, facility setup, implementation and evaluation.

Stage 2 - The Principal and Assistant Principal will coordinate the daily schedule for the Teacher and the Aide.

Stage 3 - The Curriculum Coordinator and Language Arts Coordinator will be responsible for course content and the development and utilization of a planned course of study.

Stage 4 - The Classroom Teacher will be responsible for teleteaching, student evaluation, planning and executing all lessons.

Stage 5 - The Administrative Assistant/Project Director (who is also Computer Coordinator) is responsible for staff development, inservice training, and coordinating the video productions.

Stage 6 - The librarian will assist in telecommunication and interactive video.

PROJECT TIMELINE

A. July 20, 1987 - The Superintendent and Administrative Assistant/Project Director installed the necessary hardware to meet the project goals and objectives. The Principal scheduled classes into the Hi-Tech Labs. Our students' classes are scheduled on a six-day cycle, which allows for the flexibility necessary to schedule all sophomores into the lab.

B. August 10-11, 1987 - The Administrative Assistant/Project Director conducted a staff development workshop on hardware. A Task Force composed of students, teachers, administrators, and community members was formed to help plan staff development and evaluation.

C. August 14, 1987 - Researched necessary courseware and telecommunication services.

D. August 17-18, 1987 - Inservice on the use of courseware and telecommunication services.

E. August 24, 1987 - Installed teleteaching lines to be used with the correctional institution.

F. August 26, 1987 - Inservice teleteachers and associate teachers.

G. September 1, 1987 - Inservice teacher and aide on interactive video.

H. September 9, 1987 - Pretest all students.

I. September 10, 1987 - Began teaching English class using hardware and courseware to introduce students to computer use.

J. September thru November 1987 - During the first marking period, students will work on expanding knowledge of the writing process and word processing using microcomputers networked to a laserprinter. Interactive video lessons will be created for remediation and teleteaching will begin. These objectives will continue through this year and will continue to grow in future years.

K. November 1987 - Task Force meeting to evaluate progress to date.

L. November thru December 1987 - While continuing with the initial objectives, students will be able to access commercial data bases. They will be taught how to complete a menu and a query search.

M. January 1988- All students complete a mid-term evaluation.

N. January 1988 - Task Force meets to evaluate progress.

O. January thru April 1988 - Utilizing the objectives learned to this point, the students will be able to refine cognitive and affective skills through research and creative writing. Having the freedom to write, without the worry of making mistakes, will allow for more time on task. Research verifies the fact that increased time on task increases learning.

P. April 1988 - Task Force continues to evaluate the program.

Q. April thru June 1988- Students will learn and use the desk-top publishing system. All students will complete a research report or a research paper which will be formatted and word processed. It will contain research data retrieved from commercial or public domain telecommunication data bases.

R. June 1988 - Final Exam (Post-test).

S. June 1988 - Task Force evaluation of the first year.

T. June thru August 1988 - Execute summer programs in the Hi-Tech Lab.

U. Since the project is being built into the curriculum, it will be an on-going process, with on-going staff development and evaluation. Over a period of years, all aspects of the project could expand. Teleteaching, for instance, could extend into other areas of the curriculum. Telecommunication projects such as electronic mail and bulletin board usage, plus additional data base services would also be possible. A large library of interactive video lessons will be developed for student use. All of these projects will assist in building and expanding computer/writing technology in the classroom over the next several years.

SCHOOL DISTRICT COMMITMENT

The School District is dedicated to improving instruction. As the classroom of the future and the work force of the present will require Hi-Tech skills, we are committed to preparing our students for the years ahead. This commitment has been demonstrated through the progress that we have already made.

The school district will provide an area in the library (resource room) for the interactive video system. Also provided in this area will be an Apple IIe computer and its interactive video interface. The librarian, or aide, will assist students who want to use the system.

The district will pay the expense of the telephone systems necessary for teleteaching.

The Administrative Assistant/Project Director will provide inservice programs, as well as an on-going staff development program.

At the Senior High School, a room with is 912 square feet, will be wired with the necessary equipment and house the Hi-Tech Lab.

Our school district will also provide access to commercial data bases. Over a period of three to five years, we will also phase in the following: Access Pennsylvania; Penn Link: Partnerships with local businesses (hopefully incorporate CAD); additional teleteaching projects; and an IBM Writing To Read Program in the elementary schools.

The school district will provide the class with an AppleWorks Classroom Pack, as well as additional materials.

The English Department plans to continue this program as an integral part of the curriculum. The year following the program year, an additional class may be added to work on more advanced writing skills and more detailed desk-top publishing. Since our art teacher is already expressing a strong commitment for graphic art in desk-top publishing, the second year of the program will involve the art students.

With the growing need to handle more data, telecommunication facilities will be telescoping over the next ten years, and our school district is committed to expanding this project as technology expands. We are currently evaluating Jack Taub's Education Utility (the classroom of the future).

PERSONNEL

Our district has five people who are PDE approved computer instructors. They have conducted workshops and inservices for other school districts and have recently taught a 15-hour PDE course to interested faculty members at district expense. Additionally, 90+% of our staff have attended a five hour basic computer literacy course which was also taught by these certified instructors.

The Administrative Assistant/Project Director/Computer Coordinator is well qualified in this area and has been an innovator in the field of computer implementation. He will oversee the program and will coordinate the staffing and scheduling of the Hi-Tech Lab.

The district's Language Arts Coordinator has been actively involved in implementing computer assisted instruction for disadvantaged readers. Her expertise in the teaching of the writing process will enable her to plan the implementation of the Hi-Tech Lab for the English

Department.

The Business Department Instructor who will assist in teaching students job-related word processing skills is a specialist in the field of word processing. Through frequent contact with the employers in the area, and by taking continuing education courses in word processing, her knowledge of technological advances in the field is current. With six years of experience teaching word processing at the secondary and post-secondary level, she is extremely well qualified and will be the Business Department's facilitator for implementing the project.

Additionally, a Level II computer course has been planned and will soon be offered to all district personnel. If teachers are interested in pursuing graduate level courses in the computer field, our district prepays their tuition. Many opportunities for further staff development exist and periodic reevaluation of our program allows us to continue updating the qualifications of our instructors.

EVALUATION DESIGN

The evaluation of this project will be formative as well as summative. The evaluation task force will collect data pertinent to anticipated and unanticipated outcomes. First, they will collect baseline data prior to beginning the project. A pre-test will also be used, and an immediate post-treatment instrument will be administered (final). Since task force members will represent all content areas, periodic meetings will facilitate discussion on how effectively the process is being integrated into the various curriculum. (Example: Well composed and constructed papers in social studies.)

The evaluation task composed of members of all content areas will evaluate the effects of intercurriculum writing being produced in all classes.

Incorporating input from local industries who have utilized word processing hardware, the Business Department will develop competency based standards for determining acceptable hard copy. A major international corporation has developed a pay-incentive program for word processing equipment operators. This program is used locally by three employers who frequently hire our students upon graduation. We intend to adapt this program to provide us with a means of evaluating our students' ability to meet the word processing standards required by industry. This evaluation method will also be the basis for determining a grade for the course.

The English Department will require all sophomore students to accomplish each of the stated objectives. As a result, we project that 95-100% of our students will be able to compose on the computer at the end of the 87-88 school year.

At mid-term, and at the end of the first year, students, faculty, and community members will be surveyed to ascertain the progress of our plan.

We believe the following tendencies also enhance the measurable cost effectiveness of our program. Attention span is noticeably increased as students work with the computer. Our careful selection of appropriate courseware will provide students with endless opportunities for practicing and improving writing skills, and developing and perfecting employable skills. The additional benefit of learning by doing or "enhanced learning" will be a hallmark of the Hi-Tech Lab.

COURSEWARE SELECTION

The criteria of user friendliness, ease of use, and powerful editing was essential in selection of software. The Milliken Writing Workshop, Apple Access, and AppleWorks programs were selected for inclusion in this project, and were evaluated by a committee composed of coordinators from the Business, English, Mathematics, and Computer Departments, as well as the district's curriculum coordinator. They generated recommendations from the English, Math, and Science Task Forces, previewed numerous programs on various hardware at the Regional Computer Resource Center, and solicited input from local industries, through vocational advisory committees, regarding word processing skills. In response to these recommendations, the committee used the following sources to evaluate: On-Line Searches--EPIE on Compuserve, RICE and ERIC on BRS, RCRC Recommendations, NEA Yellowbook, TESS, and numerous articles in current periodicals. Word of mouth recommendations from other teachers were received via the Apple Computer Company Northeast Region Sales Representatives.

HARDWARE SELECTION

The selection of hardware for the program used a form similar to the selection of software. The Apple II GS Network System with a LaserWriter printer was selected for inclusion in the program. As these computers will be used by students who have received no formal computer training, it was necessary to select the hardware that was most user friendly, yet very cost effective. The fact that the Apple II GS is more powerful than the Apple IIe and, is, in effect, (all add-ons considered) less expensive, is phenomenal. Members of the committee were able to receive hands-on experience using a variety of hardware and software at the local Regional Computer Resource Center. They found that Pennsylvania Higher Education Assistance Agency made an excellent choice in the Networks with the LaserWriter. The GS with its newest technology of pull down menus, icons, mouses, and windows, proved to be extremely easy to use. This is of the utmost importance, since we will use if for teaching creative writing and word processing, not just computer instruction. The amount of time spent on "how to use the computer" will be minimal. In addition to the Regional Computer Resource Center, numerous high schools and universities are adding LaserWriters to their resource centers. The George Washington High School in Philadelphia has a lab using the LaserWriter and has been using it effectively as a tool in its curriculum. The committee found article after article in numerous periodicals to confirm our findings. The AppleTalk Network with the LaserWriter printer produces quality print which is unmatched by any other hardware. The ITS interactive video interface, being used by our district, is the only interface that we could find enabling us to use teacher created tapes or commercially available tapes, and not infringe on tape copyrights. The Panasonic Commercial Grade VCR has proven to be the best quality at the stated price. The final consideration was the prompt service our district has received from both our I.U. repair service and the local Apple dealer.

PRACTICALITY AND APPROPRIATENESS

In summary, we contend that this project is exciting, workable, and very manageable in our district. With our Hi-Tech minded administration, computer literate staff, and a supportive, progressive community, this project will not only work well, but will be an ever expanding and developing project. The impact this project will have on correctional education, alone, is significant. The fact that members of our Computer and English Departments do inservice programs for the I.U. and several other districts, will enhance the growth of this concept beyond the geographic lines of our district.

REFERENCES

Donahue, Tom, "Educators Learn Computers Are A Writer's Friend Too", *Times Herald*, 1986.

Oberschelp, John A., and Zemke, Suzanne *The Writing Workshop*, Milliken, 1985.

INTEGRATION INTO K-12 CURRICULUM AND DISTRICT IMPLEMENTATION

Ruth Catalano
Chris Morton, Don Beverly
Michael P. Warner, Dennis A. Johnson

ABSTRACT: Marketing Support Services

Ruth Catalano, Director, Instructional Materials, Washington School District. Priscilla Lundberg, Facilitator of Technology, Washington School District, Phoenix, AZ

Selling technology as a "real" service is essential for technology program implementation. How do you get 1,200 teachers and 60 administrators to want to use the resources available? In the Washington District the 88/89 school year represented the seventh year of technology implementations of various modalities.

Marketing support services is a multi-level and multi-modality process. The levels deal not only with the current interest and abilities of the audience of receivers, but also with the use of technology in the curriculum and the availablity of the resources. The multi-modality process reflects the listening, receiving and accepting styles of this same audience. These levels and modalities are constantly changing.

In reviewing marketing of support services in a school district, the review must constantly shift between what should be, what is and what could be. The components of the marketing process consist of addressing these various publics. The emphasis of this service should be infused through several layers of support. The layers addressed will be staff development, external and internal communications, and documentation of delivery as part of accountability.

ABSTRACT: Evaluating School District Computer-Use: An Assessment Process Model

Chris Morton, Computer/Curriculum Coordinator of the Putnam/N. Westchester Board of Cooperative Educational Services
Don Beverly, Coordinator of Educational Technology, Lower-Hudson Regional Computer Center, Yorktown Heights, NY

This program was developed to help school districts formalize the evaluation of computer-use in their classrooms and laboratories for the development of short-and-long-term planning. The program has been piloted in a number of small, rural school districts and focuses on close involvement of all district personnel, students and parents.

Regional and state-wide analyses of school district computer-use shows that the development and planning of computer-use is haphazard, poorly coordinated, and almost never clearly monitored: curriculum design excludes technological applications which are found as "add-ons"; purchasing of hardware and software is piecemeal, and decisions are often made by individuals who have little or no understanding of the educational value of their purchases; broad-based educational objectives lead to inequitable, uncoordinated and speculative applications.

Expanding concern motivated by this typical situation lead a number of school districts to request help in the analysis of process designs to satisfy both sound educational criteria and boards of education whose questions about expensive technological installations and uses have become embarrassing. Initial analyses of extant procedures give very little satisfaction and thus this project was developed.

The presentation of the project focuses on the process that was used to gather information, analyze the information, and encapsulated the information in a form acceptable for use in planning. The discussion includes the determination of appropriate sources of usable data from faculty, administrators, students, parents, the community and board members and the methods of obtaining this data such as surveys, direct observation, interviews, and brain storming sessions. Weaknesses and strengths of technology programs are targeted with simple paradigms.

The development of the report and final recommendations are discussed, and are tied to an on-going monitoring process, the implementation of change, and the kinds of pitfalls that should be avoided. We conclude with methods of presentation of the assessment and report to the school district and its various components.

Each participant will receive a model of a report and copies of surveys and questionnaires used in the process to use and adjust as they find useful. No part of this program is a commercial product.

After the presentation participants should have some idea of the following items:

1. The need to monitor and assess all school district computer-use.
2. Items to look for in an evaluation and why they should be targeted.
3. How to develop an evaluation process.
4. Who to target in such an evaluation process.
5. How to use the evaluation data to develop recommendations.
6. How to present recommendations and work towards implementation of those recommendations.

Further help in the development of local district technology planning will be offered, and various programs available throughout the country which would help districts formalize their technology and curriculum planning will be referred to.

ABSTRACT: Aligning the Computer to the Curriculum: Model and Implementation

Michael P. Warner, Cadre Services Coordinator, DuPage-Kane Educational Service Center, Wheaton, IL
Dennis A. Johnson, Faculty Associate, Illinois State University, Normal, IL

It is possible to identify four major models of educational computer use over the past 10 years: Programming, CAI, Applications software, and Integrating the Computer into the Classroom. Each of these models had a serious shortcoming: the use of the computer was driven by hardware and software, not by a district's educational objectives.

To correct this shortcoming, a new model for educational computer use was developed. The new model, called "Aligning the Computer to the Curriculum", is based on the premise that computers in our schools should be a part of the process that delivers identified educational objectives - not a "Thing" that needs to be taught to students. In the process of aligning the computer to the curriculum, the focus is on classroom teachers and their ability to supplement educational objectives through the use of computers. Because classroom teachers are most knowledgeable about their educational objectives, they must be the ones who determine when the computer can be appropriately used in the classroom. The "Alignment' model specifies the skills classroom teachers need in order to align the computer to their curriculum.

The process of aligning the computer to the curriculum began at the Lab School for Illinois State University during the 1987 school year. Faculty members at the school identified existing curricular objectives they wished computers could supplement. Language arts and business education were the first areas in which the computer was aligned to existing curricular objectives. The presentation will discuss aspects of the "Alignment" model and will present the results of alignment at the Lab School.

TEACHER EDUCATION

Bonnie Price, Pam Kinnaman
John Ray, Marvin Peyton
Sharon Smaldino, Tina Sherrill-Range
Donald Pratt

ABSTRACT: Teacher to Teacher Training, ESTTI Style

Bonnie Price, Mentor Teacher in Technology, Whittier City School District
Pam Kinnaman, Instructional Specialist, Whittier City School District

The Elementary Summer Technology Training Institute, sponsored by the State of California in 1986 and again in 1987, provided a variety of opportunities for technological staff development in grades K-8. Through carefully orchestrated classes organized into five curricular areas and inservice training, the 400 teachers or administrators who attended the month-long residential program benefitted greatly. Two factors were especially appreciated: the careful nurturing of the group identity and the team building activities throughout the training. The inservice component was invaluable as it prepared the participants to prepare for presentations thoroughly and to modify them on the spot to meet the needs of the audience. It also prepared participants to assist in a variety of technological areas, from stand-alone computer use to interactive videodisc instruction.

At a time when the regional technology centers were being closed by the Governor's veto action, we returned to our district with a high level of energy and a commitment to help our fellow teachers in the district to become technologically literate. As participants in the Elementary Summer Technology Training Institute, we felt we were well outfitted for the tasks ahead.

Our plan for the year began with presentations to various groups --from school board members and administrators to classroom teachers and PTA presidents--to familiarize them with the current technologies and to promote the use of technology in the district. We then taught a series of classes on software and hardware at the district office after hours to interested teachers and inserviced teachers on site by request of the teachers or the principals.

The model of staff development training provided by ESTTI is an important one that should be considered by state goverments nationwide. At a time when staff development and school reform is a major issue, this kind of innovative program can halp greatly to revive interest in learning through peer training. Teachers learn best from those with the greatest amount of credibility: the in-class teachers. The effect of California's Elementary Summer Technology Training Institute and its two sister institutes, the Math and Science Technology Institute (for teachers of grades 7-12), and the English, Language Arts, Social Science, and History Summer Technology Institute (for teachers of grades 7-12) is being felt daily throughout the classrooms of California as revitalized teachers use technology integrated into the curriculum to enrich instruction.

ABSTRACT: Preservice Microcomputer Education Program

John R. Ray, Professor, and Director of Microcomputer Laboratory; and Marvin Peyton, Instructor, College of Education, University of Tennessee-Knoxville, Knoxville, TN

In 1985 the Department of Curriculum and Instruction at The University of Tennessee, Knoxville instituted a computer instruction component as an integral part of the preservice teacher education program. From a modest beginning, this component has grown in importance and in quality as its role in the preparation program for teachers has been recognized. Generally, the students have little or no prior computer knowledge. The major focus of the program is to familiarize the preservice teacher candidate with certain aspects of computer use as it relates to instruction and curriculum development.

The goals and objectives of the course are to:

Goal 1: The student will be aware of the general concepts of computing.

Objective 1: The student will know:

- Historical development of computers, number in use and current price range;
- Types of computers; and
- Functions of computers.

Objective 2: The student will understand:

- Functions and workings of these functions in a computer;
- Flow of data through the computer; and
- General concepts and terminology of computing.

Goal 2: The student will become familiar with the general uses of the computer in education.

Objective 1: The student will know/comprehend the:

- Central administrative uses of a computer;
- Instructional uses of a computer;
- Concepts and terms of educational computing; and
- Various programming learning, tutorial, Computer Managed Instruction and Computer Assisted Instruction uses of a computer

Objective 2: The student will apply/analyze the implications of the microcomputer in the school.

Goal 3: The student will analyze and apply the concepts of CAI.

Objective 1: The student will understand the:

- Definition, purposes, scope and limits of CAI;
- Underlying theories of CAI;
- Various types and formats of CAI; and
- Elemental concepts of CAI.

Objective 2: The student will apply/synthesize the understandings developed earlier into a well developed CAI program designed to teach a single concept.

Goal 4: The student will know computer software and one approach to software evaluation.

The class has been conducted for approximately 1,250 students, who have successfully completed the program. The verbal feedback has been overwhelmingly positive and the constant use of the computer lab by former students is indicative of a successful program. The only negative comment has evolved around the amount of work involved in one quarter of effort. The students, though very verbal in indicating dislike for the heavy assignment load, as a group are very positive with their newly acquired skills at the completion of the program.

Students are very aware of their acquisition of basic skills in using the microcomputer. This set of user skills, coupled with teaching skills, has permitted some students a latitude which would never have existed. These successful students are able to provide their students with programs designed specifically for their individual teaching style.

Learning the basics of microcomputer operation also is a positive outcome of the class. Many of the former students are being asked to use their skills in the schools in which they now teach.

ABSTRACT: A Computer-Based System for Adult Pre-Inservice Instruction

Dr. Sharon E. Smaldino, Assistant Professor, University of Northern Iowa, Department of Curriculum and Instruction, Cedar Falls, IA 50614
Ms. Tina Sherrill-Range, Graduate Student, University of Northern Iowa, Department of Curriculum and Instruction, Cedar Falls, IA 50614

The teacher in the classroom holds the key to educational changes and advancement for the student. Teacher inservice is a major way to improve teacher education. Teacher training efforts have taken on new meaning due to time, development of alternative learning environments, efforts to meet individual needs, informal and non-threatening teacher training experiences, a push to improve basic skills, and the exploration of new materials and media. With these ideas in mind, computer-based instruction can be utilized as a pre-inservice instructional tool, thereby making inservice training more effective and efficient.

With pre-inservice teachers will come to the inservice meeting with at least a basic background in the topics to be discussed. Computer-based instruction helps by individualizing instruction, so that the teacher can select the appropriate unit of instruction from a menu of choices for his/her individual pre-inservice learning needs. The units can be selected based on the background areas in which he/she feels deficient. Equally important, teachers can decide the time frame for accomplishing the personal objectives they wish to establish.

Teachers want to improve basic skills be being involved in inservice training. Efficient and effective use of the computer can quickly become a basic teaching tool and method for informal, non-threatening pre-inservice training. Teachers want to explore new materials and media. Pre-inservice is a good way to get teachers prepared for the group inservice that will follow.

This presentation will develop the concepts of computer-based instruction and pre-inservice training, followed by a demonstration of an example of a pre-inservice computer-based instructional program for teachers. The presentation will attempt to allow for audience participation.

ABSTRACT: Educational Computing Applications at Bloomsburg University

Donald L. Pratt, Department of Curriculum and Foundations, Bloomsburg University, Bloomsburg, PA

Elementary, Secondary, and Preschool Education majors at Bloomsburg University take a required three-hour undergraduate course entitled Educational Computing and Technology. The course features the mastery learning philosophy and practical skills applications of computers. Content includes roughly four weeks each of (a) computer literacy and software evaluation, (b) graphics programs, (c) computer lesson construction (SuperPILOT), and (d) *AppleWorks* applications.

Computer literacy includes the usual familiarity with computer parts and systems as well as exposure to interactive laserdisc. Software evaluation includes selection criteria and philosophy as well as hands-on work with software. Graphics work includes student-made pictures containing text and printouts involving *PrintShop*, *Kidwriter*, *LogoWriter*, *Koala Pad*, SuperPILOT, and *MousePaint*.

The SuperPILOT sessions emphasize principles of computer lesson and screen design, lesson planning, and lesson writing. Students move on to the authoring language (SuperPILOT) phase only after successfully outlining their lessons. This unit is one of the most creative with students producing a wide range of computer lessons which they can take with them and use in classrooms.

AppleWorks applications include making a sample letter to a parent using the word processor, a grade book using the spreadsheet, and a teacher-related database. Advanced options include making a boxed-in schedule matrix using the spreadsheet and an advanced database using its calculation features. Students are also shown other advanced educational applications which the author has developed.

A mastery learning approach is used to maximize exposure to the many computer applications and to ensure personal skill in using them. With the exception of the final exam, all evaluation is based on completion of work chosen from among 10 modules having variable units of credit. Students select the modules, the order of modules, and the number of units of credit within the modules as they desire. Grading is based on the total accumulated units during the term according to a predetermined schedule. It has been found that having three evaluation dates for accumulated work has been necessary to ensure that students do not leave all the work until the end. Students report they have to work a great deal in this course, but that they also learn much.

DATABASE APPLICATIONS IN THE CURRICULUM

Julie Poage
Graduate Teaching Fellow, University of Oregon
Software Review Editor, *The Computing Teacher*
Eugene, OR

Neal Strudler, PhD
Associate Editor, International Council for Computers in Education
Eugene, OR

ABSTRACT

The database has clearly emerged as one of the critical tools of the information age. Members of tomorrows work force will increasingly rely on their ability to access, analyze, and synthesize vast quantities of on-line data. The integration of database applications throughout various academic disciplines offers one method to help students develop some of the skill necessary to work within a complex world of information processing.

This session will be composed of demonstrations and discussion of various commercially available instructional databases. Among the software presented will be *The Irish Immigrant, The U.S. Constitution Then and Now, Sunburst's Database Series, The Power of Nation States, World Data* (a Hypercard application) and *GeoWorld*. The session will offer a critical examination of the data files, lesson guides, database software, and the hardware required to integrate these programs into the curriculum. While the session will focus on the content and quality of available programs, implementation issues will also be discussed.

FrEdBase: A DATABASE PACKAGE FOR THE CLASSROOM

Greg Butler
Miller Computer Education Centre
Sydney, New South Wales
Australia

ABSTRACT

Which came first, the educational software package or the potential educational buyer? Are the software packages that are being produced for schools expecting us to change the way we teach children? This presentation will look at the development of one educational software package that was developed outside of the market influence. It will reflect upon the use of databases in Australian schools and consider how this package, or many other database packages, may be used to develop children's skills in questioning, investigating and exploring. The software is FrEdBase, a database management package for the classroom that has been designed by and developed by a teacher.

Children learn to ask questions and receive answers at an early age; ask any parent! Questioning and gathering information are an important part of social and language development. Unfortunately by the time children reach the middle school, and teachers are desperately trying to get children to design questions and obtain appropriate answers, school has often stifled these skills. Interestingly, with the increase in access to information afforded by computers, there will be an answer to every answerable question by the year 2000 (*Jerry Pournelle, InfoWorld Nov, 1986*); providing you know how to ask the appropriate question. Obviously, the skills of questioning and obtaining appropriate answers are becoming increasing important in schools today.

The use of database packages, such as FrEdBase, as investigating tools is growing in Australian schools.

Many teachers are finding benefits in having children create databases using information collected from or related to their own world. In this way children are allowed ownership of the information, which is an important factor in promoting exploring and investigation. Using this approach the emphasis remains on children using the database package as an investigating tool to develop skills in investigating, not on developing skills in using a software package. For this to occur the software needs to be easy to use while supporting the way children investigate.

This is the arena in which FrEdBase has been developed. It is a software package designed for use in the investigating classroom by teachers and children who enjoy exploring. It allows teachers to promote investigation and provides students with a tool to play with information and present their findings in many ways.

Most importantly, FrEdBase is in its infancy and will grow with time as teachers and students use it and decide how they would like it to be.

This type of investigating environment is available to all teachers who have access to computers. For those teachers who are interested in taking the necessary steps to create an appropriate environment that will develop children's investigating skills, FrEdBase is there to support you.

EDUCATIONAL SOFTWARE FOR SCHOOLS: THE DUTCH CHOICE

Jef Moonen
Department of Education
University of Twente
The Netherlands

ABSTRACT

During the period 1984-1988 the Dutch government has stimulated the use of computers in education by executing the Informatics Stimulation Plan (INSP). A budget of 270 million guilders used to provide schools with hardware and inservice training. Although the development of educational software was also a major activity within the INSP, the resulting educational packages did not meet the expectations. Therefore, the Dutch government has decided to give a new major impulse towards the production of educational software. In this respect a new project has started called Program Development for Computers in Education.

For the next four years, a total amount of 24 million guilders is available to develop educational software for primary, secondary and vocational education. The first phase in the project is concentrated toward a search for problem areas within the curriculum of each of the chosen educational sectors, areas in which use of computers could help to overcome apparent difficulties.

During this first phase, 34 so-called white books have been produced, each of them related to one sort of problem area in the current curriculum. Ultimately, after consultation between a great number of official advisors, the Dutch Minister of Education has chosen 18 subject areas for which the project will develop educational software during 1988. The curriculum areas for primary and secondary education are given.

Curriculum Areas for Primary Education

- Elementary reading: using the computer for remedial teaching, for individualised instruction and for differentiation in speed, level of practising.
- Applied arithmetic: using the computer for practising and problem solving in an applied context (word problems).

Curriculum Areas for General Secondary Education

- Dutch spelling: remedial programs for individual use, to overcome persistent spelling difficulties within this area.
- French: an integrated language-environment for learning French, including wordprocessing and database facilities, to be used by individuals and in groups.
- English: an integrated language-environment for learning English, including wordprocessing and database facilities, to be used by individuals and in groups.
- English: thematic subjects to stimulate communication skills between students and to create quiz and test facilities to be used by students.
- Geography: database programs to provide information retrieval facilities: the database will contain numerical, textual and graphical information about developing countries.
- Mathematics: software tools to draw functions and graphs: to be used as a demonstration tool in the mathematics class as well as by individuals and groups.
- Mathematics: software to solve equations in mathematics: remedial programs to support students in solving first and second degree algebraic equations.
- Physics: measurement tools to be used in the physics laboratory for registration purposes, data collection and data processing.

Curriculum Areas for Vocational Education

The vocational sector in Dutch education has a great number of different subsectors which have been combined into three clusters: the Economic-Administrative sector, the Technical sector and the sector for Service and Health education.

Curriculum Areas for the Economic-Administrative sector

The choice was mainly based upon the circumstance that comuter use in this sector was not yet widespread. Three priorities have been chosen:

- Improving keyboard skills.
- mproving wordprocessing skills.
- Learning about administrative processes using computing facilities.

Curriculum Areas for the Technical sector

- A toolbox to support measurement, interpretation of graphs and statistical calculations within laboratory activities.
- Software to support individual practising in applied mechanics and for demonstration

purposes.
- Software to support the reading and interpretation of technical drawings.

Curriculum Areas for the sector of Service and Health education
- A hospital information system, to be used by individuals and in groups and to be applied in several subject areas of the curriculum of this sector.
- Software to support fashion design.

From the results of this first phase, it is clear that the Dutch educational community prefers to use educational software to support the learning of traditional basic skills (reading, writing, mathematics, measurement) and basic skills needed in the new information society (wordprocessing, data handling, information handling and retrieval).

By the end of the year 1988, the project will have produced a significant amount of courseware. At the same time, the project will start marketing activities to promote the use of the designs of these packages by other interest parties. For information, contact the project manager of the POCO-project: Drs. C. Nagtegaal, Education Computing Consortium, P.O. Box 217, 7500 AE Enschede, The Netherlands, tel. (31-53-892190).

References

K. van Deursen, The Introduction of Information Technology in the Dutch Educational system. In: T. Plomp, K van Deursen, J. Moonen (eds.), CAL for EUROPE, Proceedings of a Conference of the European Commission on The Development of Educational Software, Enschede, The Netherlands, 25-28 May, 1986. Amsterdam, North-Holland, 1987.

THE PLUTO NETWORK: AN EXPLORATION OF THE USE OF TELECOMMUNICATIONS IN TEACHER EDUCATION IN EUROPE

Rhys Gwyn
Centre for Information Technology and Education
Manchester Polytechnic
United Kingdom

Norman Longworth
IBM International Education Centre
LaHulpe, Belgium

ABSTRACT

The PLUTO Project, currently in its early operational stages, is a European networking project which links together teacher education institutions in a growing number of countries. The Project is directed from the Centre for Information Technology and Education at Manchester Polytechnic and is supported by IBM at both European and national levels; other sponsors are in the process of becoming involved as other institutions formally join the Project.

The Project builds on the existing collaboration between a group of experienced teacher educators all with responsibility for training teachers in the educational uses of the new technologies of information and communication (NICT). The project is a valuable opportunity to develop understanding of the new forms of teaching and learning - in this case, shared development and collaborative distance learning - made possible by the NICT.

The PLUTO Project will:
- suggest new curriculum approaches to existing subjects;
- explore and identify ways in which the use of the technologies may enhance the resources traditionally available to the classroom; and
- possibly initiate new thinking about institutional delivery systems for education.

While PLUTO is in essence a network of teacher training institutions, a small number of schools will be associated with each institution, which will participate in the network at their own levels (Primary with Primary and Secondary with Secondary). Thus the fully-developed PLUTO will be a three-tier network, under the overall supervision of the teacher training institutions, and already a number of schools are very enthusiastically involved.

DESKTOP PUBLISHING

Joseph B. Hillberry
Stephen M. Shuller
Carol H. Singletary, Esther Cottingham
Sally Butzin

ABSTRACT: School Desk-Top Publishing

Joseph B. Hillberry, Director Computer Education, Douglas School System, Ellsworth Air Force Base, SD

Schools strapped with "shoe-string" budgets can use desktop publishing in all phases of education to produce high quality documents at a minimal cost. Professional printing tools used in business for cutting costs have a place in the school environment. This presentation will focus on how students, teachers, and administrators in the Douglas School District use desktop publishing to produce the various types of printed materials required in operating a school system.

Douglas purchased a laserwriter printer and an Apple Macintosh computer to publish the school newspaper in 1984. All elements of the school population have found this computer system as a viable means for meeting their publishing needs.

Publishing a school newspaper is a challenge, but the system met this challenge and many more. Examples of awards, ballots, forms, handbooks, memos, overlays, parent information, programs, signs, teacher materials, test, tickets, and others will be demonstrated.

The laserwriter printer is ideal for little jobs that need only one or two copies. An example is when our team qualified for the regional play-offs in football and we hosted the game. Twenty complementary tickets were needed. The laserwriter was ideal for this printing job. The last minute print job that is needed so often in a school can be handled by desktop publishing.

The Douglas system uses the Apple II line of computers for students and staff. Files from these computers are converted to the Macintosh format and printed on the laserwriter.

ABSTRACT: Desktop Publishing of High School Student Publications

Stephen M. Shuller, Computer Coordinator, Scarsdale Public Schools, Scarsdale, NY 10583

Starting in September, 1987, Scarsdale High School made desktop publishing equipment available to the editors of our student publications, including a school newspaper, opinion/news magazine, literary magazine, and journals representing student clubs. We financed our equipment and software purchases with funds previously used to typeset the newspaper and our course catalog, so the project cost nothing to implement. In subsequent years, we will realize substantial savings in typesetting costs.

In addition to saving money, our desktop publishing project has enabled us to dramatically improve the appearance of publications we could not previously afford to typeset. It has also generated a great deal of enthusiasm among students not usually interested in computers (including females and humanities enthusiasts) and restored the computer center to its early '80's level of popularity (when everyone did not have better computers than ours at home!).

The project started early last spring with a feasibility study, discussions, and planning. Over the summer we purchased and installed our equipment and software, and conducted workshops for faculty and students. In September, we started laying out and typesetting our newspaper. Before we had a chance to implement our plans for involving editors of other publications, the student "grapevine" brought them into the computer center where they learned the basics of desktop publishing from other students.

Some key considerations in starting a project of this type are:

- selection of appropriate hardware and software;
- providing means for easy data transfer (since students prepare documents on a variety of computers and word processors);
- planning for changes in editing and publishing procedures to take full advantage of the new technologies; and
- providing a nurturing environment with structure enough to facilitate learning and flexibility enough to allow a student-directed "desktop publishing culture" to emerge and grow.

Our project has shown that desktop publishing is not only a cost-effective means for modernizing school publications, but also a catalyst for broadening and enriching a high school's "computer culture."

ABSTRACT: Desktop Publishing Of A School or Classroom Magazine

Carol H. Singletary, Teacher, Hughes Road Elementary, Dickinson,Tx.
Esther Cottingham, Teacher, Silbernagel Elementary,Dickinson,Tx.

The "Desktop Publishing Of A School Or Classroom Magazine" project began as the result of students' interest in having their work published in a classroom magazine. Their interest in the resulting product led to a belief that they could participate in the process.

Second graders are taught computer vocabulary and keyboarding skills which also serves to prepare them for computer work in the future. An Apple II+ is used. They are able to help with the publishing aspect of their creative products.

At the upper elementary grades students use their computer and language arts skills to edit and publish a school magazine. The students are introduced to a new computer and a new software program. The students use a MacIntosh and MacWorks. Higher level thinking skills are used in editing, layout and other production tasks.

The magazines include poetry, prose, art work, and computer generated art based on student computer programs written in both BASIC and LOGO.

The project focuses on the integration of basic computer education into the language arts curriculum and the motivation provided to students by having their works published. The curriculum development emphasized language arts skills as well as basic computer literacy.

ABSTRACT: Computers Helping Instruction and Learning Development (CHILD)

Dr. Sally Butzin, Project Manager, Florida State University, Center for Instructional Development and Services, Tallahassee, FL

Computers can motivate and empower students by providing interactivity, reinforcement, specific feedback, and individual tutoring - all factors strongly associated with academic success. But to incorporate such technology meaningfully into classroom life, teachers need viable models and support materials that can bring computers into their daily teaching repertoire.

Researchers at the Center for Instructional Development and Services at Florida State University have developed a model for restructuring elementary school environments - Project CHILD. The CHILD model proposes a new approach to using technology to improve classroom instruction. CHILD staff are creating classroom management materials to help teachers integrate computers into the classroom setting. Computer learning stations will increase student access to computers and facilitate more effective use of curriculum software.

Schools can use existing hardware and software to implement the CHILD model, aided by three system management tools under development at the Center to facilitate the process:

- A Teacher's Manual assists principals and teachers in restructuring classrooms, as well as managing the movement of students among learning stations.
- Learning Activity Guides aid teachers in planning instruction; there is a Guide for each grade level (K-5) in reading, language arts, and math.
- Student Passports for both primary (K-2) and intermediate (3-5) grade levels help students move through the learning stations independently. Passports are sent home for parent review at the end of each unit.

The CHILD management materials, plus inservice training given by the Center's CHILD development team, will provide the guidance and support needed to establish classroom curriculum centers. The CHILD model can lead today's elementary schools toward the schools of the future, where students take greater responsibility for their own learning activities - guided by both teacher and computer.

For more information about Project CHILD, please contact Dr. Sally Butzin, Project Manager, or Dr. Dennis Thorp, Associate Director, The Center for Instructional Development and Services, Florida State University, 2003 Apalachee Parkway, Tallahassee, Florida 32301; telephone (904)487-2054.

READING AND ART: K-12

Michael French
Linda Stumvoll
Dianne Tobin
Gary Greenberg

ABSTRACT: Helping Rural Elementary Teachers to Integrate Microcomputers in K-8 Reading and Language Arts: A Course Description

Dr. Michael P. French, Department of Curriculum and Instruction, College of Education, Kansas State University, Manhattan, KS

Through its many off-campus offerings, the Department of Curriculum and Instruction at Kansas State University supports rural educators in a variety of curriculum areas. In EDCI 886: Integrating Microcomputers in Reading and Language Arts, K-8, rural elementary teachers are not only being taught foundation skills in computer literacy but also strategies for intergating microcomputers into their elemenetary reading language arts programs.

Major topics which are addressed in this course include the following:

1. A thorough introduction to the use of microcomputers in reading and language arts;
2. An overview of vocabulary and comprehension instruction using microcomputer software;
3. The use of word processing--not only as a tool for teacher production but also as an instructional tool for the teaching of writing;
4. The selection and evaluation of software appropriate for use in K-8 reading and language arts;
5. Emergent literacy: Issues in microcomputer use in early (Pre-K and K) reading and writing;
6. The use of programming (Applesoft BASIC) in reading and language arts instruction; and
7. Planning for the integration of microcomputer instruction in reading and language arts--the integrated lesson plan and class project.

These topics are introduced through lecture, discussion, and a great deal of hands-on computer experience. In fact, these "labs" have been shown to be one of the most critical components of of the course.

Student evaluation in this graduate-level course is based on attendance and participation, oral and written reviews of software, completion of lab exercises, development a formal integrated lesson plan, and completion and presentation of a class project.

ABSTRACT: The Computer as an Artist's Tool, K-12

Linda K. Stumvoll, Computer Coordinator, Northport - East Northport Union Free School District, Northport, NY 11768

For the past few years the emphasis of computers in education has been on using the computer as an applications tool, yet when it comes to art education the computer has been seen by many people as a way for students to circumvent the creative process. Drawing programs have been used as electronic coloring books, with little thought being given to the actual planning and execution of the art work, and the favorite programs of many students and teachers are for creating signs and banners using canned pictures and shapes.

In the Northport - East Northport schools the art teachers have taken a different approach to computers in art. The general consensus has been that the computer can be a tool that should be introduced to art students in elementary school and continue to be available through high school Since early 1985 members of the district's art staff have been experimenting with ways to use the computer as a creative tool for artists and art students. As interest in computers grew the art teachers, along with the district's computer coordinator, herself a computer artist and former art teacher, offered inservice workshops in computer graphics. One component of these workshops were presentations by district teachers who had developed computer based lessons.

The Northport Computer Art program has developed like the trunk of a tree that is nurtured by its roots. Many individuals worked independently to develop their own unique concept of the computer as an artists tool. Once these individual ideas were brought together they melded to form the basis of a program that is now being incorporated into district's art curriculum. Among the types of projects that have been developed for the computer are:

1. Computer generated slide shows of drawings made on the computer;
2. Montages and collages composed of graphics created on the computer and printed in hard copy;
3. Digitized self portraits that have been enhanced and manipulated by using graphics software;
4. Animated videos using drawing software and a video camera;
5. Minature books designed on the Macintosh computer; and
6. Animated films created with drawing and animation software.

Since all of the district's six elementary and three junior high schools have Apple IIe labs most computer art has been created on Apples. In the high school there is a Macintosh lab where most of the computer graphics work takes place.

Currently, the only level where all students, districtwide, have a common experience in using the computer for art is in the seventh grade. Seventh grade art is required of all students and a component of the course is in using the computer to design and execute a project. The computer has also become an integral component of the eighth and ninth grade elective art courses. High school art students use computers in a class called Videography, a two period combined art/English class in which students use the Macintosh for various forms of expression. High school students also have the option of taking a course in computer graphics which is offered concurrently with Videography.

At the elementary level most computer art experiences are offered during special workshop periods where teachers have the opportunity to work with small groups of students. It is during these workshops that students learn the basics of computer graphics and begin exploring the endless possibilities of using the computer as a tool for creativity.

The project presentation at NECC '88 will include a multi-media display of projects developed by Northport art teachers for the computer. Written outlines of the projects, as well as additional information about the district, scheduling, and curriculum development, will be available in a packet to be given out during the presentation.

ABSTRACT: Technology Software for Reading Language Arts - Authorable Computer Assisted Instruction in Reading Comprehension Skills

Dianne Tobin, Ed.D., Coordinator, Technology for Educators Program, Division of Education, The Johns Hopkins University, Baltimore, MD 21218

For the past six years, and especially since 1983, a Johns Hopkins University team, composed of special educators with expertise in computer assisted instruction and technologists with a background in solving complex system problems (and a special interest in the needs of handicapped children), have worked together to overcome obstacles to the fulfillment of the promise that CAI and authoring systems have held for dramatically advancing special education. This collaboration links the University's Education Division of the School of Continuing Studies and its Applied Physics Laboratory, the latter being a division dedicated to technical research and development on important national problems. It was recognized that authoring systems have not been widely used by teachers because, if the programs were flexible enough to meet many instructional needs, they took too long or required too much expertise to program. If they were easy to use, their flexibility was limited. The challenge was to design a flexible authoring system that was easy to use and required little time to author a lesson. The first program resulting from this team effort was a user friendly authoring system for low level cognitive functioning, called Multisensory Authoring Computer System, or simply "MACS". Currently, an additional program is being developed by the same team under a grant from the Office of Special Education and Rehabilitation.

This new program will address the learning area of language arts, with particular emphasis on reading comprehension skills. It is intended for higher functioning students. Although designbed to be used for learning disabled students, it is appropriate for reading/language arts instruction for regular students as well. Currently, two programs can be demonstrated-an authoring program to develop categorization skills and one to develop analagous thinking. Techniques related to expert systems will be suggested as ways to provide appropriate and specific feedback while keeping the teacher's role simple.

ABSTRACT: Encountering Musical Issues and Making Them Your Own

Gary Greenberg, Northwestern University, School of Music, Evanston, IL

At the Northwestern University School of Music the computer is being used in the first year of a two-year Musicianship program required of all Music Majors to enhance traditional approaches to music theory and analysis and to support a compositional approach to learning about music. The activies being developed use the Macintosh computer connected via MIDI interface to a small digital synthesizer which provides students with an orchestra of fifty instruments to work with. The music notation program *Deluxe Music Construction Set* is used to present musical scores and for student analysis projects. A music environment written in the *Object LOGO* programming language supports procedural descriptions of musical material and compositional tools for students to work with. The following model is being developed as a framework for using the computer as a tool for learning about music:

- An activity at the computer provides an initial encounter with a piece of music. Instead of a seemingly arbitrary presentation of observations about the piece, the intent is to enable students to get to know the piece in their own terms and to raise musical issues and concerns that are personally meaningful and relevant.

- The computer is used to support different analytic activities on the piece and to explore musical issues that emerge from class discussion. While keyboard proficiency often stands in the way of students being able to deal effectively with a piece of music, at the computer they are able to work more like experienced musicians, comparing and testing their theoretical judgements with their actual hearing of the piece.

- Through working with the piece from different perspectives, a model evolves that captures the development of musical ideas and the structure of the piece as a whole. This model is used as the framework for a student composition. While composition has traditionally been an important way of learning about music, it is a technically demanding and difficult problem usually reserved for more advanced students. To reduce the overhead of composing and to allow students to direct their attention to the relevant musical issues, compositional tools are provided in the *Object LOGO* music environment. Students use these tools to create higher-level descriptions of the musical events they want the computer to play.

Fundamental to the approach being presented is the ability of the young musician working at the computer to address musical issues in a way that allows for the ongoing manipulation and exploration of musical issues. In this way, students are able to work more like experienced musicians, building on their own musical knowledge, than novices blindly applying terminology and following a set of rules.

ACM—SIGCUE PRESERVICE TRAINING IN EDUCATIONAL COMPUTING: FUTURE DIRECTIONS

David Moursund, *Chair*
International Council for Computers in Education
University of Oregon
Eugene, OR

ABSTRACT

During the discussions at the Phase 1 Work Conference in January 1988 it became obvious that there are many different directions that the project could take. Several areas were identified:

1. expanding on Phase 1 bu working with colleges and universities to integrate computer usage into the standard methods courses. The creation of the guidelines for the model preservice course was considered an interim measure. It must be followed up with a broader plan for integration throughout the preservice program. Guidelines might include both content to be addressed and methods for moving the college or university towards implementation.

2. repeating the process of Phase 1 (gathering of experts, writing background papers, intensive critique and publication) with a focus on specific methods courses (i.e. computer in the math classroom, computer use in the art classroom, etc.). A set of guidelines containing recommendations for both the classroom and the preservice subject area methods course could be the outcome.

3. discussion of the possible need for several seminars to discuss critical issues in computer use in education. Experts from various areas could be brought together to address current and future concerns. Some possible seminar topics are: 1) the role of artificial intelligence and expert systems, 2) the role of programming and computer science as a subject area in the secondary schools, and 3) the role of research on learning with technology.

Evaluation of Phase 1 will also be addressed. Several colleges and universities will be selected in which to test the model preservice course. Based upon this "real world" evaluation, the course may be revised prior to final publication of the guidelines.

Participants in this session will bring other concerns about the future of educational computing for discussion with the panel members and audience.

PANELISTS

Carl Berger
University of Michigan
Ann Arbor, MI

John Lawson
Towson State University
Baltimore, MD

Robert Taylor
Teachers College, Columbia Univ.
New York, NY

Molly Watt
Center for Learning Technology
Newton, MA

SPONSOR: SIGCUE

ISSUES OF EQUITABLE ACCESS AND EDUCATION: THE INDUSTRY VIEWPOINT

Antonia Stone, *Chair*
Playing To Win, Inc.
New York, NY

ABSTRACT

Industry representatives will be asked to address the question: What are the major issues of equitable access and education in computer technology, and what is your company doing to address these issues?

Respondents will comment on the answers, and pose additional questions to the industry representatives.

Time will be allowed for questions either to industry representatives or to the respondents from those attending the session.

PANELISTS

Andrea Gonzales
Apple Computer Company
Cupertino, CA

William Gaddis
Tandy/Radio Shack
Ft. Worth, TX

Mary Pat Radabaugh
IBM
Atlanta, GA

RESPONDENTS

Carol Edwards
Project Micro
Southern Coalition for Educational Equity
Atlanta, GA

Jo Sanders
Women's Action Alliance
New York, NY

SPONSOR: ICCE

TEA-NET: THE TEXAS EDUCATION AGENCY ELECTRONIC TELECOMMUNICATIONS NETWORK

Philip F. Gehring
Texas Education Agency
1701 North Congress Avenue
Austin, Texas
(512) 463-9049

Flynn Noqueira
National Information Systems
P. O. Box 2550
710 West 14th Street
Austin, Texas 78768
(512) 472-6432

ABSTRACT

For approximately the past two years (November 1985 to August 1987) the Texas Education Agency (TEA) has conducted an experimental pilot project with school districts using a statewide electronic mail, bulletin board, and teleconferencing network. Fourteen school districts and one education service center were subsidized by the Agency to participate in this experimental use of technology for communicating between and among school districts and the Agency itself. These pilot districts were funded by the Agency to ensure regular use of the network. In return for the payments the pilot sites agreed to participate in training, to examine critically the information posted on the bulletin boards, to use electronic mail, to test the local area network software being developed by the contractor, and to provide information for a study of this experiment. This paper summarizes the report presented to the State Board of Education (SBOE) Long-Range Planning Committee regarding the successes and failures in the project, the excellent and surprising feedback from the pilot districts, and the recommendations arising out of the project.

The paper also describes in some detail the capabilities, unique functions, and cost saving features of the local area network software developed by National Information Systems in cooperation with educators at the Texas Education Agency and in the pilot school districts.

I. INTRODUCTION

For approximately two years (November 1985 to August 1987) the Texas Education Agency (TEA) has conducted an experimental pilot project with school districts using a statewide electronic mail, bulletin board, and teleconferencing network. Fourteen schools districts and one education service center were subsidized by the Agency to participate in this experimental use of technology for communicating between and among school districts and the Agency itself. These pilot districts were funded by the Agency to ensure regular use of the network. In return for the payments the pilot sites agreed to participate in training, to examine critically the information posted on the bulletin boards, to use electronic mail, to test the local area network software being developed by the contractor, and to provide information for a study of this experiment. This paper summarizes the report presented to the State Board of Education (SBOE) Long-Range Planning Committee regarding the successes and failures in this project, the excellent and surprising feedback from the pilot districts, and the recommendations arising out of the project.

The purpose of the Texas Education Agency Telecommunications Pilot Project has been to study the efficiency and effectiveness of disseminating information electronically between TEA and state school districts. The project is related directly to Goal #8 of the Long-Range Plan of the State Board of Education for Texas Public Schools, which seeks a better way to communicate with Texas schools.

> "Building on public or private expertise, the state will create and implement an integrated communication system. The state will encourage the use of electronic information delivery systems."

II. BACKGROUND

The Agency contracted with National Information Systems (NIS), (i.e., THE ELECTRIC PAGES) in Austin, Texas, to provide unlimited electronic mail, bulletin board, and teleconferencing services from TEA to all districts and education service centers already using the network. The two primary reasons for selection of the NIS network was the availability of numerous other education bulletin boards already established on the system and the fact that over 250 schools districts and all 20 education service centers were then using the network. For example, when the contract was initiated, the Texas Association of School Boards (TASB), the Texas Computer Education Association (TCEA), the Junior College Association, and several other educational entities had existing electronic bulletin board and electronic mail networks on the NIS system. Subsequent to TEA's involvement with THE ELECTIC PAGES, the University Interscholastic League (UIL), the Teacher Retirement System (TRS), the Texas Association of School Administrators, and the Southwest Educational Development Laboratory (SWEDL) have also been added to the network.

The Contract

The original contract also contained several other important features. Each of the 14 pilot districts had additional campus sites included in the experiment. The total number of sites in all of the State Board of Education member areas amounted to about 48. Personnel from each district and all 20 education service centers were trained at the Texas Education Agency as well as staff in the Agency itself. The contract provided for NIS to develop local area network (LAN) communications software and established low purchase prices for Texas school districts, TEA, and NIS programmers. The contract also specified hotline telephone service and the development of inquiry software that could be used on-line to search large data bases. Finally, the contract called for a study of the pilot district experiences and a report to the State Board of Education which is the basis of this paper.

In his pilot districts report transmittal letter to the State Board of Education, Dr. William Kirby, Commissioner of Education, stated the following:

> "The pilot site experiment has been informative. Feedback from superintendents, administrators, and teachers in the pilot districts was very positive and will guide our future communication planning efforts. They generally recommend statewide implementation of electronic communications with the commersurate elimination of the distribution of information in paper form. Electronic communications is a radical departure from the standard communication and information dissemination techniques now used in school districts as well as in the Agency. However, I believe that the report implies that many schools are ready for this step forward in communications technology and efficiency and that we can reduce paperwork while increasing the flow of information in public education."

Dr. Kirby also noted that the recommendations in the report would be incorporated into a long-range for the use of technology to assist public schools in Texas. This plan is now under development and will be presented to the State Board sometime in 1988.

The Communications Highway

As stated in the objective, this project has focused on studying the effectiveness of electronic communications between TEA and school districts, as determined by the pilot site test. However, the ongoing goal of NIS and many education groups around the state has been to build an electronic "communications highway," whereby educators could easily and economically access timely data they need from many educational sources and for communicating with other schools and other educators.

THE ELECTRIC PAGES began in March 1983 with a daily legislative bill tracking program sponsored by the Texas Association of School Boards (TASB). The initial application was used by 20 school districts. Since that time, the system has grown to include 10 education groups and assocations that provide information networks to 520 school districts of the 1,086 districts in Texas.

In September 1985, TEA published an implementation plan that outlined the responsibilities, budget constraints, and process for developing an electronic communications network for schools. The plan also addressed the approach to be used to establish the pilot districts that would test and evaluate the network.

The Agency funded two hours ($30) of connect time per month per pilot site. The sites, which included 14 school districts, including 47 sites within the districts, and one education service center, used 42 percent of the connect time TEA provided. On an average, the 48 sites accessed the system 50 minutes per month.

Despite the growth of the network among school districts, acccess to the system has been limited to one or two people in each district. This is primarily because of cost (i.e., connect time charges). In effect, TEA had transferred to schools its costs for mailing information.

In order to make an electronic "communications highway" available to a greater number of educators without increasing costs, NIS designed and developed unique communication software, called The Network Kit. This software has evolved from their experience over the last four years of work with schools, education associations, and the Texas Education Agency. Effectively, The Network Kit is communication software designed by educators for educators.

III. SYSTEM OPERATION

THE ELECTRIC PAGES is a telephonic network. To access the system, a user dials into THE ELECTRIC PAGES host computer, located in Austin, using either a Texas 1-800 number or a local number. A user accesses the system with any type of microcomputer, modem, and communications software. THE ELECTRIC PAGES host computer is a Texas Instruments minicomputer with 600 megabytes of storage and 16 telephone lines.

A telephonic network can operate at either 300, 1200, or 2400 BPS (bits per second--the speed at which information is transmitted across the phone line.) THE ELECTRIC PAGES presently provides 300 and 1200 baud service with plans to initiate 2400 baud service during the fall of 1987.

In order to access the system, a user must have an ID and Password. A user is charged $24 per year for an ID. The ID includes a private mailbox that allows users to send and receive electronic mail messages to and from other users on the system.

In addition to the ID, a user is charged for the connect time to the system. The 1-800 number charge is $15 per hour, this includes long distance and computer usage. The local number is $12 per hour during the day and $6 per hour in the evenings.

Once in the system, the user can perform three primary functions:

- **Access information from private networks such as TEA-NET.** The information in the networks is stored in a bulletin board on THE ELECTRIC PAGES host computer. The information remains in a network until it is removed by a network operator. TEA-NET, like other networks on THE ELECTRIC PAGES, is a private network. Thismeans only those ID's that have been coded to access the system can do so. TEA-NET is available to all public school personnel, with the exception of students.

- **Send and receive electronic mail to and from other users.** Mail is private. No one can read a user's mail, unless the person signs on with the user's ID. Users do not have to be onnected to the system in order to receive electronic mail. Electronic mail messages are stored in the user's mailbox on the host system until the user signs on and "picks up their mail."

- **Participate in electronic conferences.** An electronic conference allows as many users as necessary to input information which can be shared and seen by all others in the conference. Conferences can be private or public.

IV. THE NETWORK KIT

THE ELECTRIC PAGES has, since its inception in March 1983, evolved as the sophistication and number of users increased. The host system has been redesigned three times in the last four years in response to user feedback.

The design of The Network Kit communication software is a result of this evolutionary process. It will allow the system to move from a low-volume, low-subscriber environment to a high-volume, high-subscriber environment. (See Appendices A & B for Charts on Low to High Volume Environments.)

The current design of THE ELECTRIC PAGES is best suited to a low-volume, low-subscriber environment. The primary reasons for this are connect charges and availability of telephone lines.

In the case of a statewide network in Texas, it would be costly and inefficient for approximately 6,000 campuses to dial into one host system for information and communications.

Furthermore, from the viewpoint of a school district, unconstrained network usage creates a variable cost that cannot be budgeted. Thus, school districts must currently limit network access to the superintendent and possibly a few other people in the district.

From the viewpoint of TEA, increasing the volume of information provided to the districts has the effect of increasing connect time and leads to similar budget problems.

The goal of a statewide education communications highway is to achieve a high-volume, high-subscriber environment so that large quantities of information may be accessed by many people in the local education agencies. The increase in volume and usage must be accomplished at a flat, reasonable cost.

To meet this objective, NIS has developed the concept of a multitier or distributed network, and developed the communication software, The Network Kit, which implements the concept.

The Network Kit has two components: InfoExchange, local host software, and InfoAccess, the user access software. (See Appendix C for Network Kit features.)

The InfoExchange program is the interface with the central host in Austin. InfoExchange receives and stores statewide data bases and serves as the electronic mail collection and distribution station for the local education agency. InfoExchange then serves all InfoAccess callers in the district through local telephone calls. Thus, information and communications from the state host are distributed throughout the state through InfoExchange.

InfoExchange is able to provide access to a large number of users within a school district with negligible variances in cost.

Broadcast Information

Increases in the volume of information received by InfoExchange will also affect operation costs. At the present, all transmissions are telephonic, thus subject to long distance tariffs.

As the statewide network evolves, it will be necessary to investigate the cost benefits of a broadcast network versus a telephonic network. In order to meet the high-volume, high-subscriber environment, the central host must broadcast information to the local host sites. Utilizing broadcast technology transfers the cost of information distribution from the variable cost telephonic network to the fixed cost broadcast technology.

The Network Kit has been designed to operate in a broadcast mode. (See Appendix D for Satellite Operations.)

Daily Operations

During the daily operations of The Network Kit, information and mail from the state host computer is automatically downloaded to InfoExchange at a locally determined time of day or night. Administrators, teachers, and students are then able to access mail and statewide information from the local InfoExchange with a local phone call.

Campuses access the information from InfoExchange in an unattended, automatic mode using InfoAccess, the user portion of the software. Campuses do not necessarily have to use InfoAccess to connect to InfoExchange. Other communication software may be used. However, without InfoAccess, the user does not have the automated features.

In addition to providing districts automatic access and dissemination of information from the state to district and campus levels, The Network Kit provides each district with its own local electronic communicaitons network. This capability will revolutionize the manner in which district offices, and campuses transmit messages and information and routinely communicate. (See Appendices E and F for an explanation of the daily operations and local communications using The Network Kit.)

Solving Existing Problems

In addition to providing a system that minimizes the cost and allows for high subscriber access, The Network Kit addresses the issue of standardization. One of the biggest problems in operating a statewide network thus far has been the issue of non-standardized user hardware and software.

Lack of operational ease and convenience are other major factors that have discouraged schools from using the system fully. With everyone using a different piece of hardware and software, support and training have been difficult.

Users, for example, have experienced problems in the area of incompatability of their word processing and communication software. Although it is more cost effective to create information off-line and upload (transfer) the document into the network, the process a user must go through is often time consuming and confusing. Moreover, the process is different with each word processing and communication software package.

InfoAccess has a built-in text editor, which means districts will be able to create documents off-line without having to use an outside word processor. Districts will still have the ability to use word processing software, but the process of transferring the materials from the word processor to InfoExchange and THE ELECTRIC PAGES will be greatly simplified and everyone will be using the same process.

Under contract with TEA, NIS is providing The Network Kit on MS-DOS machines and the Apple Macintosh computer. National Information Systems also intends to make the software available in the UNIX environment. The only difference between the two current versions is navigating through the program. With an MS-DOS system the user will use the return and arrow keys to maneuver around the screen and issue commands--with a Macintosh, the user will use the mouse.

V. PILOT SITE FEEDBACK SUMMARY

In April and May 1987, TEA and NIS personnel visited the pilot sites and talked with the superintendents as well as district, campus and the service center contacts, to determine their experiences, attitudes, and recommendations, about using an electronic network to communicate with TEA and other schools. Based on their feedback, TEA and NIS have learned the following valuable lessons regarding the operations and management of a statewide electronic network.

Implementation

- Implementing a statewide electronic network is an evolutionary process and thus takes time. It also causes major changes to occur, which, initially, cause problems.
- More comprehensive implementation planning is necessary to permit smooth transition from one medium to another.
- The system is a valuable, worthwhile tool. Schools want a more efficient communications system. They believe the electronic method is the way to improve and enhance the current system.
- TEA must continue on the course it has started. The pilot sites believe the system must be mandated in order to get all local education agencies involved and for both TEA and the local education agencies to benefit from such a system.

Operations

- The system must be easy to use in order for schools to operate it. It must allow for automatic or unattended operations.
- Districts have been reluctant to join the system because this project has been labeled a "pilot project."
- Receiving information without a signature is a concern, particularily to the superintendents.
- Making time to use the system is a problem because the pilot sites are having to operate a dual system (electronic communications and traditional mail.)
- TEA should help fund the cost of accessing information from TEA-NET.
- The location of equipment, including phone lines, must be convenient or district, campus, and service center personnel will not use an electronic network regularly.
- Personnel time and telephone lines are critical costs pilot sites incurred and should not be overlooked.
- Information overload is a major concern to the local education agencies. If districts are going to receive information from TEA electronically, they want a classification scheme so they can easily search for and download information that pertains to them (i.e., superintendent, administrator, teacher) or that requires immediate action.
- All districts are not open to teachers having access to all information from TEA before district policy has been established. Districts believe a classification scheme and indexing capability will help alleviate this problem because teachers would be able to access information relevant to them.
- Implementing a classification scheme places the responsibility on TEA to determine how information is classified.
- Schools want to be able to receive and complete TEA reports through TEA-NET using a uniform technology.
- The information on TEA-NET must be more current and more useful; otherwise, districts and campuses will not access it.
- The Agency must respond to the electronic queries from districts in a more timely manner.

- Schools want to use the system for communications between districts.

- Schools want and need more training on the use of the system.

VI. SUMMARY OF RECOMMENDATIONS

Based on the pilot site feedback, NIS offered several recommendations to the Texas Education Agency for expansion and improvement of TEA's statewide electronic communications network.

- The Agency should recommend The Network Kit (InfoExchange and InfoAccess) communication software be used by all local education agencies.

- Remove TEA-NET from a pilot status and announce to the local education agencies that TEA is moving to an electronic method of communicaitons over the next few years.

- Assist districts by defraying the cost of hardware or connect time in accessing information from TEA.

- Study the cost effectiveness as well as daily communications operations, using InfoExchange and InfoAccess in a few model districts for one year.

- Index the information in TEA-NET for ease of distribution within the local education agency hierarchy (i.e., superintendent, principal, administrator, teacher). Keywords should also be used to describe each document (i.e., social studies, math, finance).

- Design a uniform applications technology that would allow local education agencies to access and report standard TEA data.

- Direct improved use and greater use of TEA-NET by the divisions and departments in TEA. Increase the quantity and timeliness of TEA-NET information.

- Encourage the use of the system for district communications.

- Implement a statewide training program to train administrators, teachers, and students on The Network Kit.

- Conduct a test of satellite data transfer as a more cost effective method to distribute large quantities of information from TEA to the district's InfoExchange host.

VII. PROJECTED COST SAVINGS

National Information Systems also reported that the full implementation of an electronic communications system at TEA, and within the local education agencies would reduce the overall cost of communications. An electronic communications system works best in parallel with existing communications systems. Implementation of an electronic communications facility will reduce the present usage of mail, reproduction, printing, and long distance telephone systems as routine communications migrate to the electronic delivery systems. As the role of the electronic communications system matures, the cost of electronic communications will increase while the cost for all other communications systems will decrease. The result is a net savings to education in the State of Texas by approximately $2.8 million per year.

III. SUMMARY

The Texas Education Agency has completed a successful two year experiment in electronic communications. Pilot school districts expressed satisfaction with the electronic network and the number of districts using the system has more than doubled.

Powerful and easy to use communications software has been developed and tested by NIS and educators. This software reduces the cost of network operation by distributing system capabilities to the local district and resolves numerous operational problems associated with non-standard communications software.

The contract with National Information Systems has been renewed for fiscal year 1988. The cost for operating the TEA-NET has been incorporated as a standard line item in the Agency budget. Two model districts are being subsidized to fully implement The Network Kit communications software on all campuses. The purpose of the model districts is to learn about the problems and costs of a totally electronic communication system within a school district. The Commissioner of Education has written to all school districts advising them that the experimental pilot project is completed and encouraging them to begin to expand their use of electronic communications.

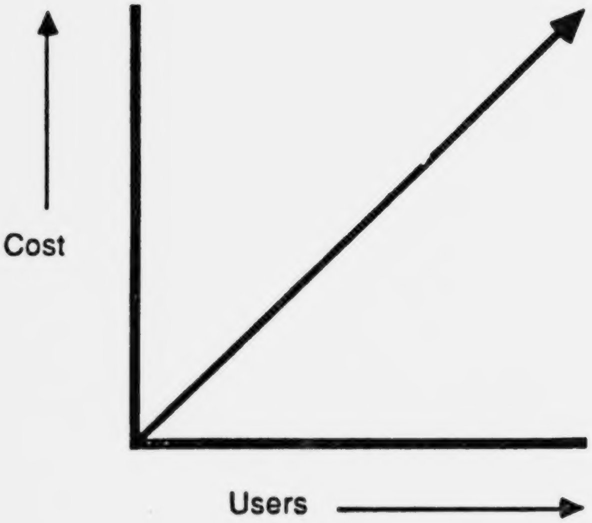
Orientation training is being offered to all Agency employees to ensure that more of the valuable information available at TEA is communicated electronically to school districts.

The next steps will result from the direction provided by the State Board of Education when they act on the recommendations in the Long-Range Plan for the Use of Technology in Education in Texas.

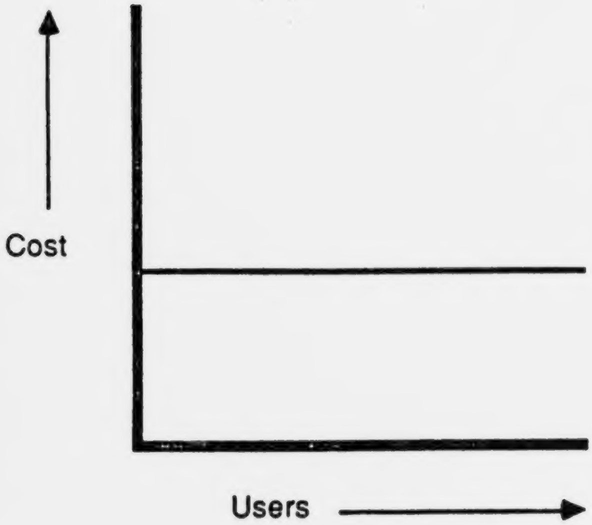
Table 1.1
Comparison of Annual Operations Costs

		Present Cost	Projected Cost	Net Savings
1.	TEA Annual Communications	$1,195,300	$ 871,370	$ 323,930
2.	TEA ELECTRIC PAGES Cost	12,000	15,000	(3,000)
3.	LEA Annual Communications	3,405,000	340,500	3,064,500
4.	LEA ELECTRIC PAGES Cost	120,000	817,200	(697,200)
	TOTAL	$4,732,300	$2,044,070	$2,688,230

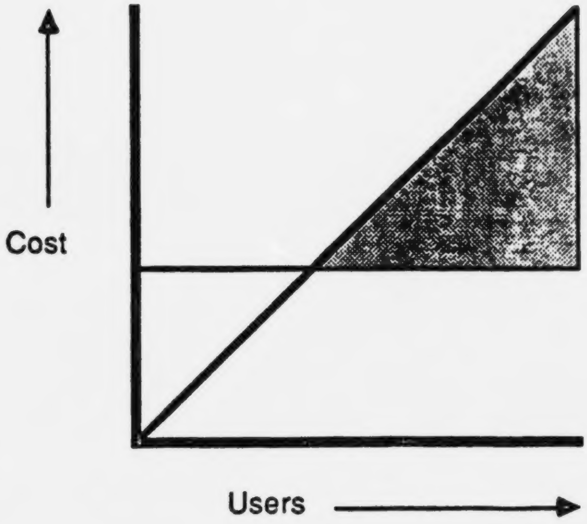
Communications Cost for a School District as Number of Users Increases



Central Host Communications Model



Distributed Host Communications Model

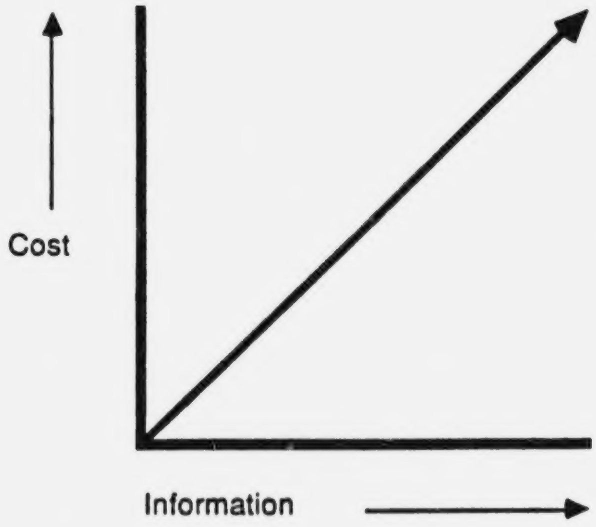


Comparison of the Two Host Models

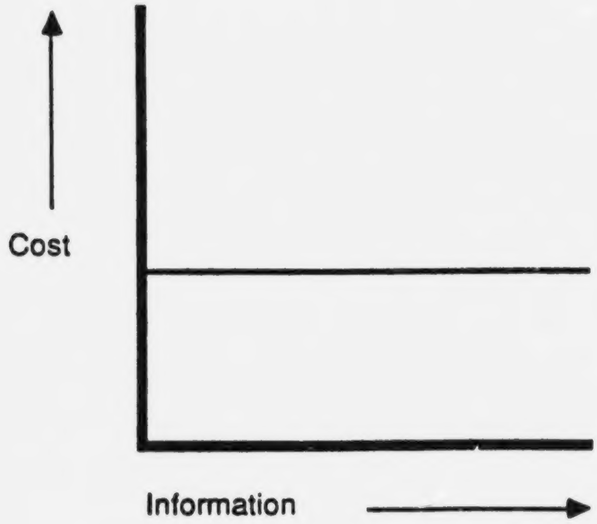
Note: Breakeven comes at three users. Shaded area represents savings when district has more than three users.

APPENDIX B

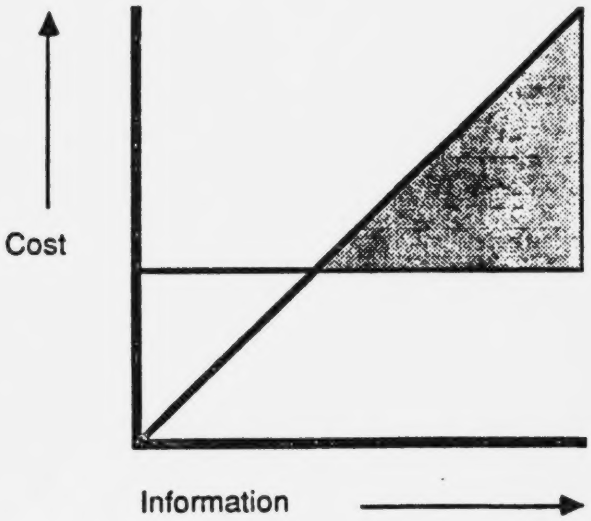
Communications Cost for a School District as Volume of Information Increases



Telephonic Based Communications Model

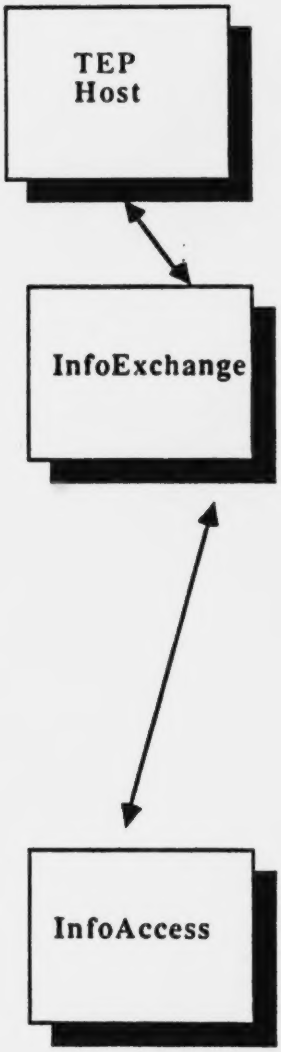


Satellite Based Communications Model



Comparison of Transmission Models

Note: Breakeven comes at ten minutes per day. The shaded area represents savings of broadcast vs. telephonic information reception.



THE ELECTRIC PAGES™

Network Kit

InfoExchange Features:

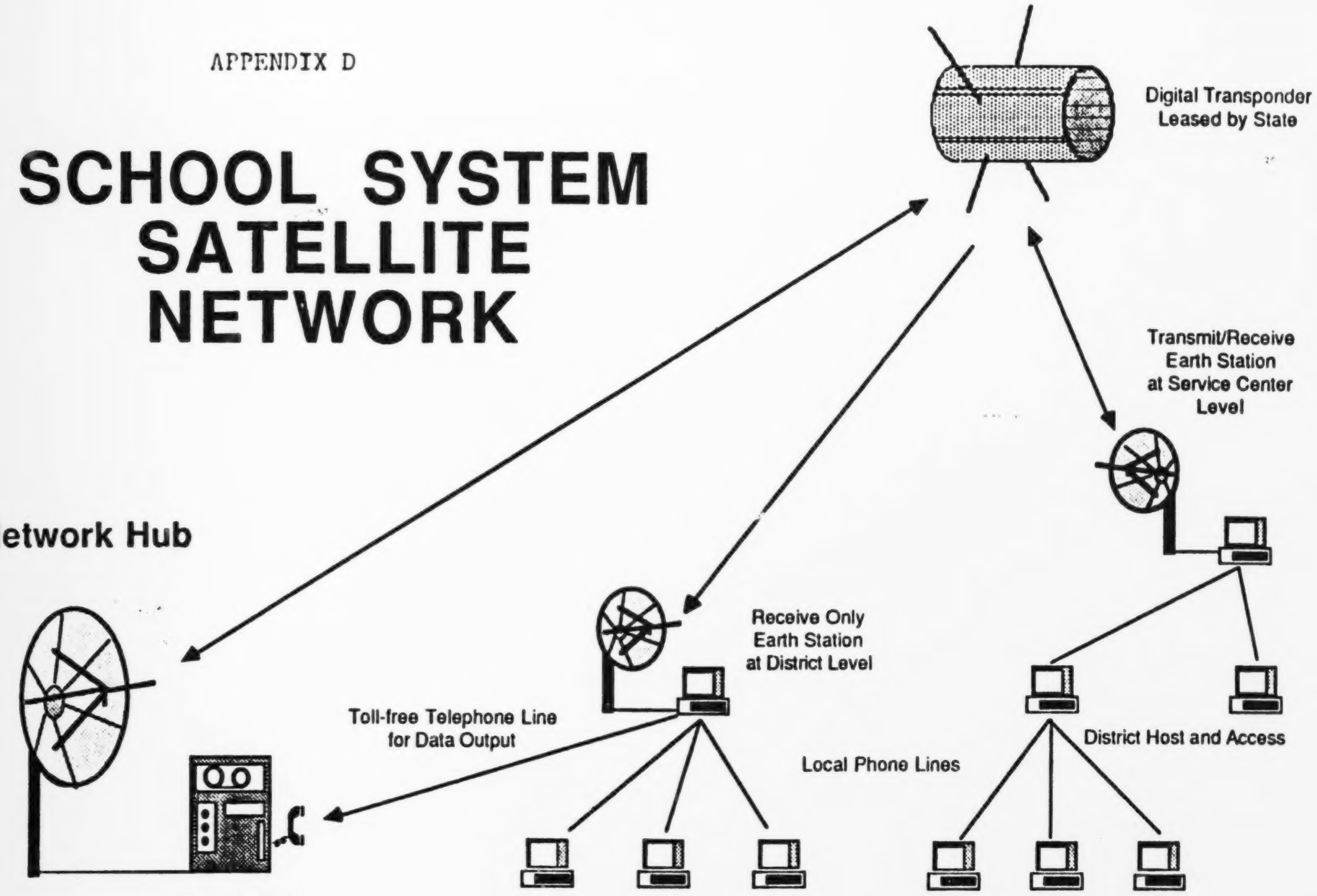
- Multiuser Service
- Electronic Mail & Conferencing
- Data Base Facilities Allowing Organization And Storage Of Files
- Unattended Operations
- Standard Text And Xmodem File Transfer Protocols And THE ELECTRIC PAGES Error Free Protocol
- User Access Via THE ELECTRIC PAGES Infoaccess Module As Well As Other Terminal Emulator Products
- Local Electronic Networking Operations

InfoAccess Features

- Unattended Operations
- On-board Electronic Mail System Capable Of Supporting Multiple Users
- On-board Electronic Mail Distribution Lists And Directories
- On-board Text Editor
- File Transfer Protocols: Standard Text, Xmodem, THE ELECTRIC PAGES Error Free Protocol

APPENDIX D

SCHOOL SYSTEM SATELLITE NETWORK



Digital Transponder Leased by State

Transmit/Receive Earth Station at Service Center Level

Network Hub

Receive Only Earth Station at District Level

Toll-free Telephone Line for Data Output

Local Phone Lines

District Host and Access

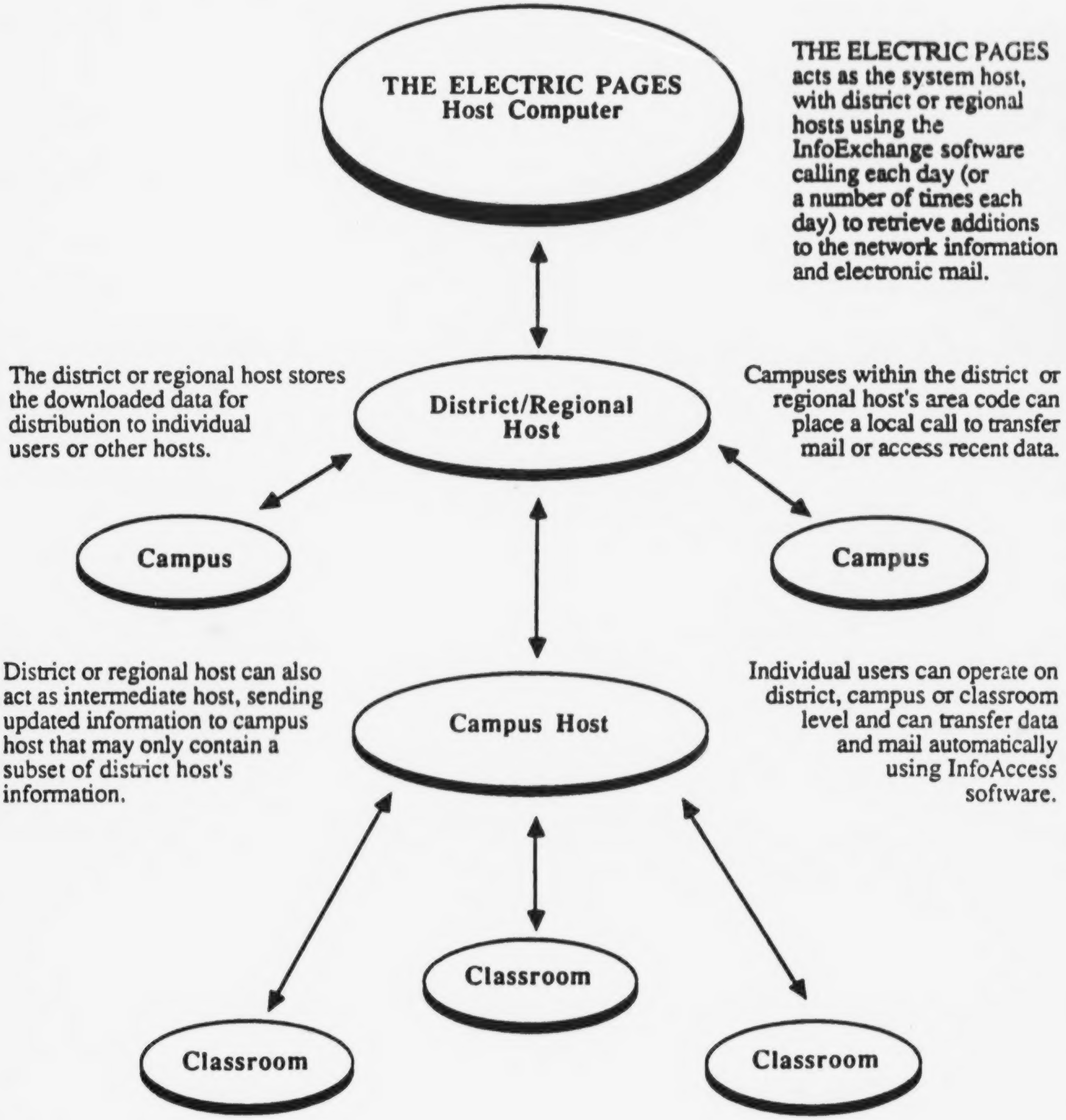
Network Host

Campus Access Points

Campus Access Points

APPENDIX

Daily Operation of The Network Kit™


THE ELECTRIC PAGES
Host Computer
District/Regional Host
Campus
Campus
Campus Host
Classroom
Classroom
Classroom

THE ELECTRIC PAGES acts as the system host, with district or regional hosts using the InfoExchange software calling each day (or a number of times each day) to retrieve additions to the network information and electronic mail.

The district or regional host stores the downloaded data for distribution to individual users or other hosts.

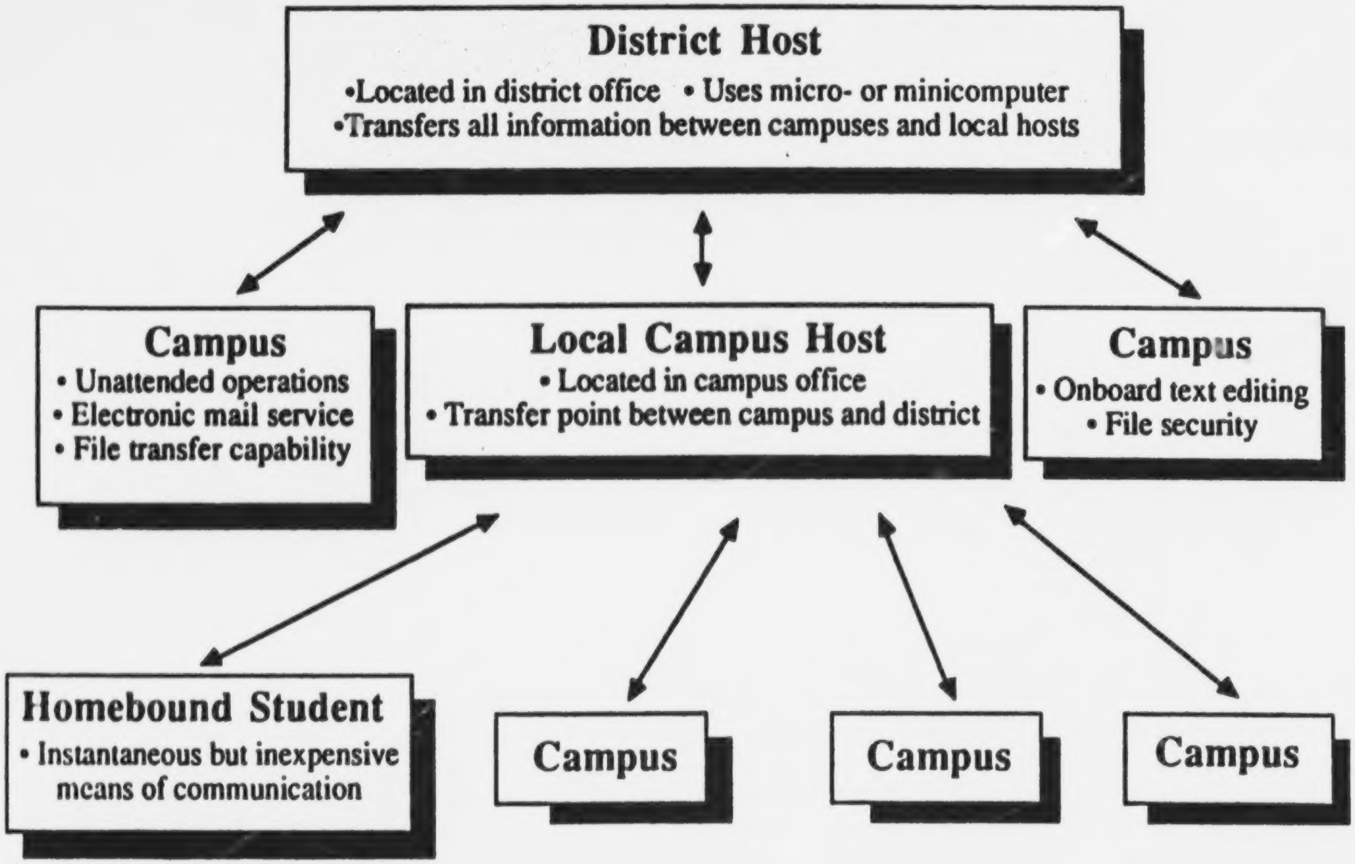
Campuses within the district or regional host's area code can place a local call to transfer mail or access recent data.

District or regional host can also act as intermediate host, sending updated information to campus host that may only contain a subset of district host's information.

Individual users can operate on district, campus or classroom level and can transfer data and mail automatically using InfoAccess software.

APPENDIX F

Local Communications with The Network™



District Host
- Located in district office
- Uses micro- or minicomputer
- Transfers all information between campuses and local hosts

Campus
- Unattended operations
- Electronic mail service
- File transfer capability

Local Campus Host
- Located in campus office
- Transfer point between campus and district

Campus
- Onboard text editing
- File security

Homebound Student
- Instantaneous but inexpensive means of communication

Campus

Campus

Campus

IS SOFTWARE LICENSING A SENSIBLE ALTERNATIVE FOR SCHOOLS?

Robert M. Caldwell
Vice President, Academic Skills Division
Learning Technologies, Inc.
4255 LBJ Freeway, Suite 131
Dallas, Texas 75244
(214) 385-2351

Lamont Lockwood
P.O. Box 7566
Dallas, Texas 75209-7566
(214) 827-7581

ABSTRACT

As computer use in schools steadily increases, educators are beginning to show considerable interest in licensing as a possible alternative to purchasing large numbers of single software titles. The reasons for this interest are fairly clear. First, schools naturally want volume discounts similar to those they receive when they purchase large quantities of audio-visual equipment and textbooks. Second, schools realize that substantial discounts on software have been passed on to businesses through licensing and wonder if this arrangement will transfer to meet the needs of the educational community as well. Finally, licensing can ameliorate the sensitive issue of illegal copying which is becoming an source of increasing tension between schools and publishers.

The issues which surround licensing for schools, however, are complex and, for the most part, difficult to define. Schools do not have the same historical or functional analogs for using software that businesses do. In addition, publishers of educational software lack evidence that they can price and manage software licenses in a manner that allows both the publisher and the schools to benefit. It is unclear, therefore, whether the traditional concept of licensing can be effectively utilized in schools. Hybrid approaches, however, do warrant further study and consideration.

In the paragraphs that follow, we will define some of the licensing schemes that are available to schools and discuss some of the advantages and disadvantages of each. We will also discuss some of the issues which surround licensing and make suggestions for both schools and publishers about how to address these issues.

A SHORT HISTORY OF LICENSING

The whole notion of licensing software originated in an era when most large companies commonly purchased *Site Licenses* for software when they bought or leased large mainframe or minicomputer systems. These licenses were a standard element in the mainframe environment for a number of reasons:

1. In most cases, software was included as an integral part of the system. Software was developed for a specific computer system and usually for a specific purpose which was tied to the system itself. If a customer wanted a particular application, they generally bought or leased the entire hardware and software system that addressed that application.
2. Most early computer systems were complex configurations which required specialized support and expertise to operate them properly. Problems, which occurred frequently, were assumed to be hardware or software related. Therefore, support and the continual upgrade and modification of the software was usually included in one maintenance contract/license and was performed by the system provider.
3. Early software applications were difficult if not impossible to use without consistent support from the developer. Therefore, this intimate customer/provider relationship prevented unauthorized copying as an inexpensive means of acquiring software.

In short, it was difficult to use a mainframe or minicomputer system *without* the support of the software provider. Therefore, acquiring a license for a particular site which also housed a large computer system was the only real alternative anyone had.

ADVANTAGES OF LICENSING FOR BUSINESS

For most large companies today, the proliferation of personal computers has made licensing a wise business decision. The cost and convenience of licensing as opposed to the purchase of individual packages of software programs for hundreds, or even thousands, of employees seems to have significant advantages for most businesses. Some of the more important of these advantages are as follows:

1. Software can be licensed at a lower per copy price than purchasing it.
2. Users can control the distribution of the software within a company without fear of the liability illegal copying presents.
3. The software can be modified legally or upgraded to meet the user's specific need at the site where it is to be used.
4. The software can be distributed on Local Area Networks or on mainframe networks to meet the needs of multiple users without fear of violating

copyright laws.

5. Training can be planned and managed through the software provider.

6. Software can be standardized across applications (e.g. everyone uses the same word processing software) to improve efficiency within a company.

To date, most of the licensing agreements that have been negotiated between businesses and software publishers have worked to the mutual advantage of both parties. The user pays a significant sum up front and then pays a yearly maintenance fee for unlimited use of the software. This fee usually represents a significant savings over what the company would have paid if they had purchased individual programs through normal retail channels. The advantages for publishers include receiving their money up front without the cost of duplicating or packaging the programs and, in many cases, eliminating the costs of selling the programs through dealers or distributors. In addition, since neither party has to worry about the pirating issue with a license, it all seems like an agreeable business arrangement. Whether similar arrangements can be made to benefit schools depends largely on the type of license that might be appropriate to the school environment and the conditions under which these licenses are to be implemented.

TYPES OF SOFTWARE LICENSES

Most discussions of software licensing are clouded by a lack of recognition that there are several types of licensing agreements available. Each type has certain advantages and disadvantages for a given user. The following is a discussion of the types of licensing agreements that are currently available and of some of the advantages and disadvantages of each:

1. Single Version Licensing

Most discussions on this subject seem to imply that licensing per se is fundamentally different from purchasing a single software program. In reality, however, *all* software is "licensed". When users purchase a software program, what they are really *buying* is the media on which the program is delivered, not the software program itself. The publisher still owns the software.

Some companies such as the Microsoft Corporation of Seattle, Washington make their licensing agreement clear to each user by printing it on each package. The envelope that contains the Microsoft *WORD*^{tm} word processing program, for example, states, "Microsoft, as licensor, grants you, the licensee, a nonexclusive right to use...this...software on a single computer...at a single location. If the single computer...is a multiuser system, the license covers all users on that single system."

By opening the sealed disk package which encloses the program diskette, "...you are agreeing to become bound by the terms of this agreement, which includes the software license..."

Microsoft is quite serious about this license agreement and if there are any doubts about that, you only need to read a little further. They state:

> THIS AGREEMENT CONSTITUTES THE COMPLETE AGREEMENT BETWEEN YOU AND MICROSOFT CORPORATION. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT OPEN THE DISK PACKAGE. PROMPTLY RETURN THE UNOPENED DISK PACKAGE...TO THE PLACE WHERE YOU OBTAINED THEM FOR A FULL REFUND.

Now, one quickly gets the impression that Microsoft is not joking about their licensing agreement. They make its terms absolutely clear to the user. Most other companies do not state their policies quite as emphatically, but close examination of their policies would reveal that their intent is very similar to Microsoft's. For the purpose of our discussion here, then, the terms of the Microsoft agreement can be presented as the generally accepted standard for a "Single Version License".

Obviously, the Single Version License is a relatively new concept in software distribution that was devised to meet legal requirements for protecting copyrights and to cope with the problem of illegal duplication, an issue that first reared its ugly head with the advent of the personal computer. Despite the fact that a Federal court in Louisiana recently declared that a "shrinkwrap" license of this type is not a binding contract, most all single versions of software purchased today carry a Single Version License with it (Swartz, 1987).

2. Site License

The most common type of license purchased by companies is the *Site License*. The Site License is designed to allow unlimited duplication of a software program by a single customer at some specific location. This location can be a company, a university campus, a government agency, or a specific school building. Users violate their license when software programs show up at sites other than the designated one unless the agreement provides for this contingency.

The primary advantage of the Site License is cost savings and user convenience. There are, however, a number of disadvantages to Site Licensing which can apply to either schools or businesses. Some of those disadvantages are as follows:

A. Administering the license. The Site Licensing agreement that most companies purchase permits an unlimited number of copies to be made for use at a designated site. When the license is terminated, however, many software publishers require that all copies be returned or that the programs be erased from the diskettes. Under either of those circumstances, locating all duplicated copies can become a logistical nightmare. Therefore, if a user is to properly comply with the license, a system must be put in place at the site to monitor the number of copies that are made and to identify which individuals are using them.

B. Cost. In addition to the cost of the license, the user must also incur the cost of buying the diskettes and duplicating the software and documentation. This cost can be high, particularly for school districts with limited resources and large student populations. This is a major factor for many companies and school districts that choose to negotiate *Bulk Purchase Agreements* (discussed below) rather than Site Licensing agreements.

C. Flexibility. Large-scale site licensing can be expensive and for that reason represents a fairly large financial commitment on the part of the user to a particular software program. Once a license has been negotiated, the users trained, and the applications installed, it is difficult to switch to other programs that offer more features or a lower price. Therefore, when a user licenses a specific program, they should take care to evaluate it thoroughly so that they are sure that it is free of bugs, that it meets the needs of the application for which it is intended, and that the publisher has a proven track record of providing upgrades and enhancements.

There should also be careful consideration of whether technological advances will soon make the application obsolete. To insure some degree of flexibility, many licenses include maintenance and upgrade provisions so that new versions can be made available without additional cost to the licensee.

3. Bulk Purchase Agreement

For all intents and purposes, the *Bulk Purchase Agreement* constitutes nothing more than a collection of Single Version Licenses. This type of agreement is by definition a license only in the sense that a customer is purchasing a collection of individual licenses at volume prices and nothing more. Under the terms of a bulk purchase agreement, the user purchases a large quantity of software at a reduced price and the publisher supplies the diskettes, documentation, and support needed to install that software. In many cases, this can be cheaper and easier for most school districts than any other form of licensing. The primary advantage of the Bulk Purchase Agreement is that it can carry with it significant cost savings if there is sufficient volume in the sale for the publisher.

Bulk purchase agreements can, however, have two distinct disadvantages. The first is that a user pretty much standardizes on a single piece of software once the purchase is made. The user makes the investment in the software and cannot terminate the agreement to avoid future payments or amortize the fee as they can with a Site License. The second disadvantage is that the Bulk Purchase Agreement prohibits unlimited copying of the software which is, of course, the primary advantage of most other site or multi-copy licensing agreements.

4. Geographically Limited License

The *Geographically Limited License* has great applicability to school districts, states, government agencies, and other organizations that are spread over a specific geographic area. The terms of this type of license usually provide for unlimited copies to be made within defined geographic boundaries.

A consortium of school districts in the state of Georgia recently entered into such a licensing agreement with a number of software publishers to make specific educational software programs available to students in participating school systems throughout the state. Under the terms of the license, each school district pays a fee to the consortium. The Georgia Department of Education then copies the programs and makes them available to any school that is a member of the consortium, providing, of course, that they were not beyond the geographic boundaries of Georgia.

The primary advantage of this type of license is the cost savings it offers the users, particularly where large numbers of learners might take advantage of the software. In addition it could mean that the sponsoring agency might save on the need to purchase a network, or it can mean that networks can be used more effectively.

A disadvantage of a Geographically Limited License, as with any Site License, is that the expense of duplicating the diskettes can be high unless the programs are distributed by contract on a high volume basis, over an electronic network, or over a satellite transmission.

5. Limited Duplication License

In situations where only a small number of programs are needed but the user wants to take advantage of the price break associated with licensing, a *Limited Duplication License* offers an attractive solution. The main advantages of this type of license is that it is easier to keep track of the number of copies made and it keeps the cost of disk duplication down. Its only real disadvantage is that it might be easier to simply seek a bulk purchase agreement rather than negotiate a separate license for the software. However, the circumstances under which the license will be used are the primary determinate of what is best in this situation.

In summary, it can be said that certain types of licensing can offer four (4) major advantages:

1. Costs of acquiring software can be reduced and amortized over a extended period.
2. Users are free of concern over the issue of illegal copying.
3. Licensing offers a certain degree of flexibility. The license can be terminated so that new, updated software can be obtained.
4. Users can achieve standardization. By standardizing on a single software title, operations can become more efficient and more uniform.

The type of license that a user might want or the type that might be best for a given situation is largely dependent upon the circumstances under which the software is going to be used and the benefits that might be derived from a particular type of license. Many businesses have been able to use some of these licensing agreements to their advantage. However, whether these same advantages will carry over to schools is an issue which at present is still far from being resolved.

LICENSING ISSUES FACING SCHOOLS

Licensing could potentially offer for schools many of the same advantages that it does for business. Schools, for example, want quantity discounts on software purchases. This is currently being achieved through bulk purchase agreements but could be achieved in other ways as well. Licensing could also help resolve the problem of illegal copying which is currently rampant in schools. Finally, well-conceived licensing agreements could provide schools with the same flexibility that businesses enjoy by receiving updates, new versions or completely new products in a planned manner.

There are, however, a number of major stumbling blocks to be removed before workable licensing agreements are put into place in schools. The first obstacle is that publishers who provide educational software are confused about how to establish reasonable fees for a license of any type that will satisfy a school district's desire to save money but at the same time compensate for revenues which might be lost from potential sales of single version programs. A number of important factors are significant to this issue if it is to be resolved in the near future. Many of those factors are related to differences which exist between publishers who sell exclusively to businesses and those who focus primarily on the school market. Some of those differences are as follows:

1. Price Margins

The majority of software distributed into the business market is negotiated on considerably higher price points than educational software which typically retails at a range between $29.95 or lower and $59.95. Few educational programs even near the $295-495 range that many business packages do. This tremendous difference in price is not an insignificant factor in trying to establish a fee that will make licensing advantageous to publishers.

To date, few publishers have really found an effective formula to use to price a software license. Some have tried to structure licensing agreements on the same model they use to determine the price for networked software. The industry-standard pricing scheme that seems to have taken hold for networked software is to charge users three times the retail price. Therefore, if a program retails at $39.95, the price for a networked version is approximately $120. This price entitles the user to unlimited use of that program at a single site for an unlimited period of time. What the networked version becomes in reality, then, is a limited site license, or what is sometimes called a "limited use" license.

In pricing networked software, publishers have tried to determine a price that will account for revenue lost from schools not buying multiple copies of a certain program. The issue with licensing, then, is similar: What price would be fair if one unprotected copy of a program was licensed to a school district for unlimited duplication within that district?

One factor that might be significant in that pricing decision is the number of students who will use the program over some specified period of time. Some publishers, therefore, are considering basing their licensing fees on some sort of a per pupil cost. The formulas for calculating these costs, however, are still fairly imprecise.

2. Standardization

Publishers who have turned to their colleagues in business for help with pricing have had little luck. The reason for this, as stated earlier, is that the differences between what business needs from a license and what schools need are fairly significant. To begin with, an important motivation for most businesses wanting to license software is so that they can standardize their operations on a particular piece of software such as a word processing program.

Schools, on the other hand, are less concerned about standardizing, at least for now. Some school districts have begun to standardize the objectives in their computer curricula, usually because of directives from state education agencies or because of membership in various state or regional consortia. Perhaps, when that standardization becomes more widespread, software programs that meet certain specific objectives will become candidates for licensing in the same way that some business application programs have.

A further problem for schools is that whatever standardization might occur will be restricted to specific grade levels. Software appropriate for senior high school business classes will not be appropriate for elementary reading classes. Schools, therefore, will rarely have the numbers of users who can standardize on one piece of software at any one time that businesses do. And even if the software is used each year for new students, rapid change in software and hardware developments will quickly make the software obsolete.

A final issue related to standardization of software is competition. Because businesses want to standardize, this increases the competition for publishers of business software to provide price breaks as sales incentives so that their products will be adopted by a single company. To date, school publishers have not faced that issue to the same extent because the number of functionally similar products available for education are not as great as in the business market. Publishers of word processing software, for example, must compete with as many as 300 other programs; certainly, any type of pricing scheme that will give one publisher an advantage over another will be considered carefully by users, assuming, of course, that the features of the various programs are fairly similar.

Publishers of educational software, on the other hand, have to a large extent avoided this problem by purposely not developing functionally competing products. Instead, they have tried to compete by introducing new and innovative products that offer diversified applications that differ significantly from the competition. Unlike producers of business applications, once a program becomes an established "hit," there are few, if any, imitators. However, as more and more software products are developed, it is only a matter of time before this competitive syndrome that has infected business software will become manifest in the education market as well.

Furthermore, having a "hit" program presents yet another dilemma for the software publisher. Most products are "hits" because they have a high degree of utility for the user. Programs such as *Print Shop* and *Newsroom* have enjoyed enormous success in both the school and retail markets for this reason. But, as mentioned above, there have been surprisingly few, if any, imitators of these two products. Therefore, why should the publishers of either of these products want to license the programs when sales are excellent and virtually no competition exists? Each publisher would actually lose revenue if they were to license these products. The only possible incentive for the publishers to grant a license would be as a convenience to avoid the issue of illegal copying.

3. Numbers of Computers

Businesses license software because they have hundreds and possibly thousands of computers on which they want to install various types of software. Except for large school systems, most schools do not have large numbers of computers and, particularly, large numbers of the same brand of computers. Schools who have large investments in Apples, for example, might also have IBM, Commodore, Tandy or Atari as well. Publishers who license products, therefore, should be prepared to make product available for a variety of machines, thus adding to the cost of developing the product.

4. Support

With most licenses, support is typically included in part of the fee. What constitutes "support," however, can be interpreted broadly. Often businesses want software modified to meet various operational standards or to accommodate various operating systems. For large software developers, technical support of this type presents few problems to the technical staff which they maintain. Many educational publishers, however, contract for software development or acquire it, making modification beyond what is delivered in the package difficult. Support of this type for educational publishers, then, becomes difficult to deliver in a licensing agreement and must be factored into the pricing if such support is to be provided.

Support can include other factors also. The following represent only two:

A. Documentation. A software license may allow the licensee to make unlimited copies of the software. However, some question exists about whether documentation is provided by the publisher for each duplicated copy of the program or whether it is the responsibility of the user to duplicate it. Price breaks can be established either way, but it is an issue to be considered by both publishers and schools.

B. Training. Many software programs require extensive training before they can be installed and used. Software used on networks serves as a good example. The question to be resolved is who will do the training if there is any? A dealer? A sales rep? How will costs for his/her services be factored into the fee?

5. Dealers and Distributors

Most software licenses that businesses purchase have been acquired through direct negotiation with the publisher. Since the life-blood of most education lines is the dealer and distributor, how will they be compensated for negotiating licenses? Will publishers allow dealers and distributors to negotiate licenses or will all licenses be negotiated directly with the publisher? These are important questions to be answered.

6. Duration

Another factor to be considered in establishing the licensing fee is the duration of the license. Will the fee cover use of the software for one year? Two? Five? Should the cost be lower for each year the user agrees to use the program like a lease for retail space in a shopping mall, or should it be the same regardless of the period for which the software is used? At least one agreement with which we are familiar specifies a renewable fee for a period of five years. At the end of that period, the software license is paid up and the user then owns the software.

In summary, it would seem that there are a number of problems to be solved before schools can take advantage of licensing in the same way that businesses do. These problems may not be totally unsolvable, however, and there might be some immediate resolution to the issues mentioned above.

One solution could be found in schools exploring ways to establish critical mass to justify wide-spread use of various software programs and to find the financial resources to purchase the licenses. This could be done by schools forming regional consortia or cooperatives that would share resources as schools in the state of Georgia have done. Most states already have mechanisms set up for this type of resource sharing through agencies such as regional service centers, technical centers, and resource centers. Consortia, then, would function in the same way that large corporations do to take advantage of the benefits offered by licensing.

A similar alternative would be for large city school districts to act as the licensee but share the license with smaller districts within a metropolitan area.

Even without consortia, however, school districts can take advantage of licensing with certain kinds of publishers. The following are a few examples of the types of publishers who can compete favorably in negotiating licenses:

1. Publishers with large numbers of titles or with software series.

The Minnesota Educational Computing Company set a precedent for licensing several years ago by selling institutional memberships to schools throughout the United States. For a set fee, schools can choose any of over 300 software titles. Each title comes with a backup diskette and an 800 number is available for support. Other publishers have also built up large numbers of software titles or have developed series and could easily follow the lead of MECC in setting up similar types of licensing agreements.

2. Publishers who want to compete for market share.

In many ways, licensing can be nothing more for some publishers than a way to promote their product line. These publishers can compete in the same way that business software publishers do by offering a special incentive to use their product. Licensing can be way to introduce the product and build user familiarity.

SUMMARY

Obviously, many of the problems surrounding the licensing of software to schools need further definition before adequate solutions can be found. In the meantime, however, software licensing offers enough promise that schools should consider the advantages carefully. Some of those advantages are

1. Cost Savings

Whether schools can realize cost savings from licensing is entirely dependent upon whether publishers of educational software can find pricing structures that make sense. Clearly, some publishers are in a better position to offer creative pricing schemes than others. It is only a matter of time before one of them takes the first bold steps in this direction.

2. Convenience

Probably the best incentive for resolving some of the issues surrounding licensing is that it can possibly take the industry a long way toward solving the knotty problem of software piracy.

3. Flexibility

Provided that profitable pricing schemes can be created, licensing can have the advantage to schools of providing them with not only the original software product but also with any updates, new versions, or companion products. In addition, licensing can also have the advantage of serving to test software titles if a short term license can be negotiated.

4. Standardization

As schools focus more narrowly on standardizing computer curricula, there will be a greater need for acquiring large numbers of individual software titles. The precedent for this has already been established by states such as Texas and California that mandate state-adopted textbooks and are moving in the direction of recommended software lists.

In short, licensing, not unlike the industry itself, is still in its formative stages. As the need for various types of software evolves, so will pricing structures and applications that lend themselves to the licensing as a workable alternative.

[*NOTE*: The authors would like to thank Karen Billings and Sally Bowman for their help and advice in preparing this manuscript.]

REFERENCES

Swartz, H. "Louisiana Bayou," *DEC Professional*, August, 1987, 110-11.

Word is a registered trademark of the Microsoft Corporation.

MICROCOMPUTERS FOR TEACHER-PARENT COMMUNICATION: EXPLORING THEIR POTENTIAL EFFECTIVENESS FOR ELEMENTARY CLASSROOM TEACHERS

Donald W. Bullock
2805 N. Wanda Avenue
Simi Valley, CA 93063
(805) 584-1496

THE PROBLEM

Teachers are being pressured to communicate more effectively with parents and also use time more efficiently. Present methods of teacher-parent communication are often ineffective and time consuming. Much of this work can be accomplished with the assistance of a microcomputer.

This paper will explore the effectiveness of using microcomputers for word processing and data management for the purpose of teacher-parent communication. It is a formal attempt to make others aware of some possibilities that computers offer them.

A REVIEW OF THE LITERATURE

Problem Established

At a time when national, state, and local attention is being drawn toward education the pressure on teachers is increasing. Reports at all levels are stating that teachers need to teach more effectively, use time more efficiently, and involve parents in their child's education.

The United States Department of Education (Bennett, 1986) has published a series of research findings regarding effective education.

> Parents are their children's first and most influential teachers. What parents do to help their children learn is more important to academic success than how well-off the family is *(p. 7)*.
>
> Parental involvement helps children learn more effectively. Teachers who are successful at involving parents in their children's schoolwork are successful because they work at it *(p. 19)*.

The report of the National Commission on Excellence in Education, A Nation at Risk: The Imperative for Educational Reform, stated that more time should be spent in teaching and learning.

> 1. The time available for learning should be expanded through better classroom management and organization of the school day *(p. 29)*.
>
> 2. Administrative burdens on the teacher and related intrusions into the school day should be reduced to add time for teaching and learning *(p. 30)*.
>
> Parents "bear a responsibility to participate actively" in their child's education *(p. 35)*.

The California Commission on the Teaching Profession produced its own report, Who Will Teach Our Children?, in November of 1985 (Commons). An increase of expanded teacher responsibilities and the complexity of the work were listed as problems. Marlyn Russel Bittle, president of the California Teachers Association, stated in her column in the association's Action newspaper that one of the major goals of the commission's report was that "Teachers must have time to teach. They must be free of administrative paperwork" *(p. 2)*.

School districts throughout the nation have been involved in their own analysis of the condition of education. My school district in Simi Valley, California has developed a set of goals for school improvement based on the many reports on education. Board Resolution 38/84 was intended to be a guideline for effective schools. Four of the resolution's points deal directly with effective use of teacher time and teacher-parent communication.

1. Develop a school-to-home communication that requires frequent reports to parents from school and each individual teacher to include information about school policies, teacher expectations, and details about each class in which the student is assigned.
2. Establish a process for contacting parents as soon as possible if a student is in danger of failing or has been involved in serious or repeated incidents of misconduct.
3. Analyze and adjust school procedures that instructional time is directed toward maximum improvement of instruction.
4. Identify techniques to increase parent understanding and support of the educational program and implement agreements with parents to increase parental commitment to the education of their children as a joint venture of the school and individual parents.

No doubt there are many other such resolutions in school districts across the nation, with more to follow. It is clear that the management of teacher time and teacher-parent communication are important factors in creating more effective schools.

Possible Solution

If the Simi Valley Resolution and those of other districts are to be carried out to their full potential there must be some improvement in the techniques used by teachers to manage and communicate information. Time involved in doing paperwork must be used more effectively. Teachers will need the tools to accomplish this task. A microcomputer for the teacher is a logical choice for further study.

According to the National Task Force on Educational Technology's report, Transforming American Education: Reducing the Risk to the Nation (1986),

> Teachers in the classroom today can begin to gain familiarity with computer-based technology by learning word processing and other productivity skills that make their task easier. They should be assisted in reorganizing their work so that recordkeeping can be relegated to the computer, thereby freeing more time for interaction with students.

The use of computers to reduce teacher paperwork was echoed in the National Governor's Association's Educational Technology Task Force report which states that programs worth considering for financial assistance "Provide teachers with hardware and software to reduce paperwork" (Knight, 1986, p. 7). The California Commission on Improving the Teaching Profession's report, Who Will Teach Our Children?, states that technology holds great promise for improving instruction by increasing the productivity of teachers. It further relates the uses of computers in business with the tasks involved in education.

Word Processing

Word processing has been used by many educators for several years. I have yet to find any studies or reports on how teachers are using word processing or its effectiveness in teacher-parent communication. The few articles I have found that mention word processing as a tool for teachers do not go beyond stating that it is a useful tool (Lankewwicz, 1984; Beekman, 1985).

To understand the possible effectiveness of word processing for my teacher-parent communication I began to research its uses in business. Almost all sources discussed word processing form a business point of view (Liskin, 1984; Beekman, 1985; Byrne, 1985; Roth, 1985; Towsend, 1985; Badget, 1986; Ditlea; 1986; Honan, 1986; O'Malley, 1986).

Data Base Management Systems

Businesses also use data base management systems extensively. The only common use of data base management systems that I could find being used for teacher record keeping in classrooms was computer gradebooks. Many teachers report success with these programs (Davis, 1982; Edwards, 1983; Wiggers, 1983; Burrows, 1984; Beekman, 1985; Nelson, 1985; Sky, 1985; Carlson, 1987). For this paper I chose not to duplicate their work.

In business data base management system programs are used as electronic files. Information such as inventory, personnel records, and client records are stored using these programs. The information contained in data base management systems is often merged with reports and letters created by a word processor (Coudal, 1982; Cronin, 1984; Miller, 1984; White, 1984).

Merging Data Base with Word Processing

From the very beginning of my work with word processing I wanted to be able to create form letters to be used in the classroom. Communications from Reader's Digest, to millions of people, were more personal than the ones I sent to thirty parents.

According to my research mail merge was possible with microcomputers. (Byrne, 1985; Roth; 1985; Townsend, 1985; Rubin, 1986; Zeicheck, 1986). Bell (1985) documented several specific cases where mail merge is used. An example from the field of education is that of Deborah Zemek. She is a data systems typist at the University of Connecticut. Zemek uses mail merge for the university's bulk mailings. According to Bell,

> Software Publishing's popular "pfs" series makes the process of merging and printing data from pfs:file, a data base, into pfs:write word processor a painless and relatively simple task *(p. 69)*.

Bell concludes that when one wants to communicate the same general message to a number of recipients, mail merge can be the key to creating the "personalized" form letter. "It's a powerful word processing tool that can help create a perception that you and your company operate on a person-to-person plateau" (p. 75).

Spelling Checker Adds Further Advantages

I am, admittedly, a terrible speller. Spelling and typographical errors are always a problem when I communicate through type-written documents. The use of a spelling checker or dictionary program certainly helps to solve the problem. (Garvey, 1984; O'Malley, 1985; Ditela, 1987).

METHOD OF RESEARCH

The research for this paper is exploratory in nature. It was determined by the review of literature that the use of microcomputers for written teacher-parent communication in elementary classrooms is an unexplored area. After reviewing the literature I decided to set up a limited field research project to test my ideas. I designed a new approach to creating teacher-parent written communication.

Subjects

Four classrooms were selected at a private school in the greater Los Angeles area. Two third grade and two fifth grade classrooms were selected. The two experimental group teachers volunteered for the research while the other teachers, at the same grade levels in the school, became my control group.

Task

Our task was simply to attempt to improve teacher-parent communication with the assistance of computers along with word processing, data base, and spelling checker software. Also of importance was efficient use of teacher time. We started with very little knowledge about the hardware or software that we used to implement the project. Learning to use it effectively was therefore part of our task.

Instrument of Measure

Information was sought from the parents and teachers involved in the project. I developed a simple survey and questionnaire to solicit parent opinions on the teacher's communication. Teacher input was obtained in a much more informal manner. They were free in expressing their opinions in our discussions about the project.

Materials

When this project was first conceived I had no means to carry it out. The materials necessary - hardware and software - were too expensive for one person to supply. I applied for an individual grant from a computer manufacturer. The grant was awarded in March of 1985.

By November of that year the granting institution supplied three complete MS-DOS computer systems with 256K, dual disk drives, color monitors, and daisy wheel printers. The data base and word processing software was also supplied with the hardware. After some research on spelling checkers I bought one to supplement the grant. Hardware and most software was limited only to products manufactured and published by the granting institution. That institution also placed a $10,000 retail value limit on the grant. Software chosen for the project included pfs:write, pfs:file, and Webster's NewWorld Spelling Checker by Simon and Schuster.

Procedures

During the first half of the 1985-1986 school year none of the teachers selected to participate in the experimental portion of this project had any knowledge that they would be participants. They all communicated with parents using their normal methods. I selected the four participants in December of 1985.

At mid-year a parent opinion survey was conducted. I sent the survey to all parents in all four classrooms. No mention was made of the use of computers or any specific details of the research.

Control group teachers maintained their normal procedures for teacher-parent communications. As much as possible in close working conditions they did not know what the experimental group teachers were doing.

I worked with the experimental teachers as they created teacher-parent communications with the computer materials. We concentrated on improving quality and individualization of the written letters and notifications that these teachers sent to parents. I continued to experiment with the materials in my own teacher-parent communications and shared my techniques with these two teachers.

Both of the experimental group teachers personalized all their teacher-parent communication during the semester. The personalization was accomplished by merging letters (Appendix A), which had been checked for spelling mistakes, with a student- parent data base (Appendix B). After merging, it is impossible to tell whether the text came from the word processor or the data base (Appendix C).

The same survey that was used at mid-year was sent again to all parents. Parents in the experimental group were asked to fill out an additional questionnaire. These questions were related to their opinions of the teacher-parent communication generated by the computers.

FINDINGS

Materials

The hardware was adequate for the task. MS-DOS proved to be a little complicated, but the computers performed well. While creating excellent printouts, the daisy wheel printers were very slow. They also limited our expansion into using graphics oriented programs. Dual disk drives proved to be necessary. The teachers were still confused with disk swapping. A hard disk drive would have solved this problem, but it would have also added its own complications.

Software Publishing's pfs:write and pfs:file were excellent choices for this project. I could not, at the time I was selecting software, have found easier software to use for my task. The ease at which data from pfs:file can be incorporated into word processed documents created with pfs:write made this project successful.

Survey and Questionnaire

Parents responded that they felt the communication produced with the computer materials was an improvement over traditional teacher-parent communication. The survey results showed that parents felt they received more information from the teachers during the project. Parents perceived the form letters to be letters sent only to them individually not as communication sent to all parents.

It is evident from the questionnaire results that parents in the experimental group saw a difference in the communication between the first and second half of the school year. The vast majority of parents in the experimental group reported that they and their children received important information in the teacher's letters.

Parent comments about the teacher-parent communication were overwhelmingly in favor of the computer generated letters.

> The letters are great. They add a personal touch with the teacher. It opens a line of communication, especially when something is wrong. Thank you for the extra time to do them.
>
> We appreciate [the teacher's name] effort to communicate with us through individual letters about our child's progress at school.
>
> The letter we received yesterday made me teary-eyed! [Student name] has been so fortunate to have [teacher name] for his teacher. A little praise goes a long way and these reports, to us, were 100% praise. Keep up the great work in striving to learn what parents need and want.

While this information is not as important as teacher reaction to the project it does indicate that the objective of improved teacher-parent communication resulted from our efforts.

Teacher Reaction

The experimental group teachers started reporting results even before any formal evaluation was conducted. They reported a dramatic increase of return notes thanking them for the letters as well as other suportive comments. An increase in requests for conferences was also noted. The teachers noticed that parents who received negative comments about their child also responded favorably to the letters. These parents expressed appreciation for the teacher's efforts to inform them of the student's problems in school.

The teachers observed that they were able to communicate more information, more often, to all parents. The computer materials allowed the teachers to send out individual letters to all parents on a regular basis. The composition time required for each letter was not much longer than that required for sending a duplicated letter to the parents. Adding individual student comments to the data base did require additional time. This process took between one to two hours per letter. This seemed like a long time to us until we realized that we were reaching all the parents in the same amount of time traditionally required to communicate with only a few. The teachers stated that the comments they received as a result of the letters made the time worth while.

Both experimental group teachers expressed a sense of achievement in improving teacher-parent communication. When asked if they planned to use computers in a similar manner during the next school year both responded in the affirmative.

Researcher's Observations

Creation of personalized form letters was more difficult than I had expected. The letters had to be general enough to be appropriate for the whole class, but they also had to allow for specific information on individual students. Both experimental group teachers had years of experience in writing letters to the entire class as well as letters to individual parents. Combining both into one letter proved to be a challenge. Both of them met the challenge very well.

DISCUSSION

It is clear from the results of the research that teacher-parent communication can be improved using microcomputers. Time saving has not been established.

Educational Implications

This research did not determine if the use of microcomputers will be helpful in improving teacher-parent communication for all elementary teachers. The teachers in this study were already very effective in communicating with parents. Teachers who are not presently adept at teacher-parent communication or have no desire to improve that communication will not be more sucessful if they use computers. There has to be a commitment on the part of teachers to learning to use the materials as well as a desire to improve.

There are no clear-cut solutions to the problem of teacher-parent communication in elementary classrooms. This paper only demonstrates one possible solution. The suggested solution requires dedicated teachers who are willing to try something new. It is impossible to improve teacher-parent communication or any other aspect of education without an attempt to try new solutions.

Suggested Future Research

Due to the exploratory nature of this research I have some suggestions for future research. I hope that this research will encourage others to explore the uses of microcomputers for teacher-parent communication. Any future research on this topic should include a much larger sample. Such a change is necessary to give statistical credibility to the research. An entire school might be the next step. It is also recommended that the research include a more detailed accounting of the communication sent home by teachers. A compare and contrast analysis of the communications would be helpful.

Such studies must also use improved instrumentation for data gathering. While my instruments proved adequate to the exploratory nature of the research they did not provide enough information for a statistically based research project. This research made no attempt, for example, to measure improvement in student attitudes or progress. A more detailed account of teacher time must also be included.

CONCLUSIONS

Most importantly, I have discovered that the microcomputer generated teacher-parent communication has increased parent understanding and support for the teachers efforts. Parents and teachers have reported that these communications are more informative than those used prior to using the computer. They also reported that the communications are more frequent. Parents can easily be contacted when the student is having academic or behavioral problems. Parents were more willing to help with the student's education. The goals of the Simi Valley Unified School District related to teacher-parent communication for the improvement of schools have been met. Only the question of time saving has been left unanswered. This question can only be answered through further research.

It is clear from the literature search that businesses, large and small, have learned to use computers effectively in communication. They have been able to do so with increased productivity as well. Perhaps, as exhibited in this paper, communication and productivity may be increased for the classroom teacher by studying the application of microcomputers in businesses. Data base management and word processing, as they are used in business, were proven to be effective in communication with parents in two elementary classrooms. The research indicates that even greater success is possible. It also indicates that using computers for communication with parents provides a more professional image for teachers.

BIBLIOGRAPHY

An Open Letter to America on Schools, Students, and Tomorrow. National Education Association.

Badget, T. (1986, October). Revitalizing the small business. *Personal Computing*, pp. 135-143.

Beekman, G. (1985). The microcomputer as teacher's aide. *Proceedings of the Fourth Annual Extending the Human Mind: Computers in Education Summer Conference* (pp. 277-283). Eugene, Oregon: University of Oregon.

Bell, J. (1985, December). Multiple mailings for impact. *Personal Computing*, pp. 67-75.

Bennett, W. J. (1986). What works: Research about teaching and learning. Washington D.C.: United States Department of Education.

Bittle, M. R. (1985) Bouquets to the commission on the teaching profession. *Action*, 24(3), p. 2.

Burrows, T. & Burrows, S. (1984). It's the end of the marking period-And there goes my weekend! *The Computing Teacher*, 11(6), 31-32.

Byrne, R. B. (1985, September). The word on word processing. *Personal Computing*, pp. 25-29.

Carlson K. W. (1987, February). Grading the gradebooks. *InCider*, pp. 67-75.

Centron, M. (1985). Schools of the future: How American business and education can cooperate to save our schools. New York: McGraw Hill.

Commons, D. L.; Chair (1985, November). Who Will Teach Our Children? Sacramento: California Commission on the Teaching Profession.

Cook, R. (1985, January). What's wrong with word processors. *Popular Computing*, p. 26.

Coudal, E. (1982, August). Managers computerize to organize with DBM. *Personal Computing*, pp. 112-119 and 137-138.

Cronin, B. (1984, June). Information management. *Popular Computing*, pp. 104.

Dalton, R. (1985a, September). Software for managers. *Popular Computing*, pp. 22-30.

Dalton, R. (1985b, November). Specialized writing tools. *Popular Computing*, pp. 21-27.

Davis, L. H. (1982, August). Color Computer gradebook. 80-U.S., pp. 22-32.

Ditlea, S. (1986, October). Word processing. *Personal Computing*, pp. 73-77.

Ditlea, S. (1987a, March). How well do you spell? *Personal Computing*, pp. 91-103.

Edwards, J. (1983, November). The electronic gradebook. *Teaching, Learning, Computing*, pp. 84-87.

Eiser, L. (1987, February). Help! I'm drowning in paperwork! *Classroom Computer Learning*, pp. 26-35.

Fisher, G. (1985). Future directions in computing. *CUE Newsletter*, 8(1), p. 5.

Garvy, I. (1984, November). Spelling checkers: Can they actually teach spelling? *Classroom Computer Learning*, pp. 62-65.

Gurmundsen, R. & Glenn, A. D. (1984). Computer gradebooks: Implications for teachers. *The Computing Teacher*, 12(2), pp. 13-15.

Hively, W. (1986, May). Grade book programs pass the test. *Electronic Education*, pp. 18-19.

Honan, P. (1986, October). Captains of computing: America's top 500 CEOs benefit from personal computing. *Personal Computing*, pp. 131-133.

James, B. (1985). Computer applications in the classroom. *Proceedings of the Fourth Annual Extending the Human Mind: Computers in Education Summer Conference* (pp.267-270). Eugene, Oregon: University of Oregon.

Knight, A. (1986). Summary of the NGA's 1991 report on education. *The Computing Teacher*, 14(3), pp. 6-7.

Lankewicz, L. (1984, September). Word processing-Uses for teachers. *Electronic Learning*, p. 22.

Lindelow, J. (1984). How to implement the electronic office. International Council for Computers in Education SIG Bulletin, 1(3), pp.4-7.

Liskin, M. (1984, October). New tools for professionals: Word processing for the office. *Popular Computing*, 76-79.

Maremaa, T. (1984, October). Computerizing the professions. *Popular Computing*, p. 75.

Miller, M. J. & Goley, G. F. (1984, June). Making sense of database software. *Popular Computing*, pp. 106-109.

National Commission on Excellence in Education (1983). A Nation At Risk: The Imperative for Educational Reform (Stock No. 065-000-00177-2). Washington DC: U.S. Government Printing Office.

National Task Force on Educational Technology (1986). Transforming American education: Reducing the risk to the nation. *T.H.E. Journal*, 14(1), pp. 58-67.

Nelson, L. R. (1985). Computer gradebooks: A better deal for students. International Council for Computers in Education SIG Bulletin, 2(2), 27-30.

Norris, W. C. (1985). Improving education through technological innovation. *T.H.E. Journal*, 12(10), pp.65-68.

O'Malley, C. (1986, July). Write faster: Speedier ways to get the word out. *Personal Computing*, pp. 71-77.

Pogrow, S. (1985). State-of-the-art in administration software. *Proceedings of the Fourth Annual Extending the Mind: Computers in Education Summer Conference* (pp. 271-276). Eugene, Oregon: University of Oregon.

Pournelle, J. (1984, October). If I had a hammer...Our computing tools force us to rethink our questions. *Popular Computing*, pp. 60-62.

Pournelle, J. (1985, January). Information management. *Popular Computing*, pp. 59-62.

Robbins, O. (1985). Grading programs: What they should do. *CUE Bulletin*, 7(5) p. 8.

Roth, P. J. (1985). Using the pfs family: File, write, report. Indianapolis: Que.

Rubin, C. A. (1986, April). Pushing your filer to the limit. *Personal Computing*, pp. 69-77.

Simi Valley Unified School District resolution No. 38/84 (1984). Simi Valley, CA: Board of Education.

Sky, N. C. (1985). Classroom management tools: Grade books and progress reports. *Proceedings of the Fourth Annual Extending the Human Mind: Computers in Education Summer Conference* (pp. 271-276). Eugene, Oregon: University of Oregon.

Spencer, C. (1986, August). Enhancing your data base reports. *Personal Computing*, pp. 60-67.

Squires, D. A. et. al. (1983). Effective schools and classrooms: A research-based perspective. Alexandria, VA: Association for Supervision and Curriculum Development.

Task Force on Education for Economic Growth (1983). Action for excellence. Washington D.C.: Education Commission of the States.

Teacher's Tools--Flexible grading for students (1987, Sample Issue). AppleWorks Exclusive Reference, p. 16. San Diego: Q-mar Group.

Townsend, C. (1985). *Pfs: Software made easy*. Berkley: Osborne McGraw-Hill.

Troutman, J. G. (1986, August). Making the grade with Appleworks. *InCider*, pp. 63-67.

Urschel, W. (1985, March). Data base management software. *Personal Computing*, pp. 129-132.

White, R. (1984, June). Pfs: File and report. *Popular Computing*, pp. 155-163.

Wiggers, T. B. (1983, November). Student gradebook and records management system for the Apple II. *Educational Computer*, pp. 39-41.

Zeichick, A. L. (1986, August). DeskMerge: Mail-merge for DeskMate. *PC Companion*, pp. 13-22.

Donald W. Bullock

Elementary Teacher
Mentor: Computer Technology

Knolls Elementary
6334 Katherine Rd.
Simi Valley, CA 93063
(805) 526 -3612

Parents
Address
City, CA *Zip Code*

March 12, 1987

Dear *Parents*,

The attached grade sheets show the grades that *First Name(T)* has earned thus far this quarter. As you can see *HS(T)* *Comment2(T)*. If you have any questions about the grades please feel free to contact me.

The school year is still going very well. The subject of Math has become our biggest problem area. I have attached a recent Math Quiz. This was a quiz designed to see if the students were retaining their math skills. *First Name(T)* *Comment3(T)*. The Review Quiz will be given again on March 18th. Please help *First Name(T)* study the things *HS(T)* missed.

During the next few weeks all of the students will be videotaping their commercials. This project has always been a favorite of many students. It combines many of the disciplines we have in school into one project. *First Name(T)* *Comment3(T)*. I hope that you plan to come to Open House on April 29th. The student's commercials will be shown in our room as part of Open House. It is always a fun time for all involved.

All of the students are also involved in using the computers to edit their stories. I am amazed at the progress some students are showing. In the past students had to copy over a whole story when there were only a few simple mistakes. Often students copied their stories, mistakes and all or they didn't make any attempt at editing their stories. With the word processor program on the computers editing has become fun for *First Name*.

Thank you for your support.

Sincerely,

Donald W. Bullock

APPENDIX A - FORM LETTER BEFORE MERGING

LAST NAME: Ship FIRST NAME: Clipper
ID#: 103458 M/F: M BIRTH DATE: 2-25-77
PARENTS: Mr. and Mrs. Ship
PRESENT GRADE: 5 PRESENT TEACHER: Mr. Bullock
ROOM NUMBER: 21 NEXT YEAR'S GRADE: 6
READING: high BEHAVIOR: low
MATH: high LANGUAGE: middle
SPELLING: middle SOC. STUD: middle
SCIENCE: middle
SPECIAL PROBLEMS: no RETAINED: no
HEARING: VISION:
SPEECH: HEALTH:
SEPARATE FROM: Maggie Basset
PARENT REQUEST: no
COMMENT1: Clipper has difficulty proofreading. He is a good writer, but doesn't review his work. His math skills are excellent. Clipper is an average student in all other areas.

HS: he HSC: He HH: his HHC: His HIMHER: him HIMHERC: Him
ADDRESS: 4297 Warf Street
CITY: Seashore ZIP CODE: 90000

COMMENT2: is a model student. He understands that while drills may be monotonous, they are useful in acquiring skills and habits. Clipper is thoughtful about his duties and responsibilities and does them promptly. Some improvement is still necessary in his his proofreading of writing assignments in language arts. The exercises that he is writing with the word processor should help
COMMENT3: only had difficulty with subtraction

COMMENT4: is ahead of most students. His project looks great so far

COMMENT5:

Donald W. Bullock

Elementary Teacher
Mentor: Computer Technology

Knolls Elementary
6334 Katherine Rd.
Simi Valley, CA 93063
(805) 526 -3612

Mr. and Mrs. Ship
4297 Warf Street
Seashore, CA 90000

March 12, 1987

Dear Mr. and Mrs. Ship,

The attached grade sheets show the grades that Clipper has earned thus far this quarter. As you can see he is a model student. He understands that while drills may be monotonous, they are useful in acquiring skills and habits. Clipper is thoughtful about his duties and responsibilities and does them promptly. Some improvement is still necessary in his his proofreading of writing assignments in language arts. The exercises that he is writing with the word processor should help. If you have any questions about the grades please feel free to contact me.

The school year is still going very well. The subject of Math has become our biggest problem area. I have attached a recent Math Quiz. This was a quiz designed to see if the students were retaining their math skills. Clipper only had difficulty with subtraction. The Review Quiz will be given again on March 18th. Please help Clipper study the things he missed.

During the next few weeks all of the students will be videotaping their commercials. This project has always been a favorite of many students. It combines many of the disciplines we have in school into one project. Clipper is ahead of most students. His project looks great so far. I hope that you plan to come to Open House on April 29th. The student's commercials will be shown in our room as part of Open House. It is always a fun time for all involved.

All of the students are also involved in using the computers to edit their stories. I am amazed at the progress some students are showing. In the past students had to copy over a whole story when there were only a few simple mistakes. Often students copied their stories, mistakes and all or they didn't make any attempt at editing their stories. With the word processor program on the computers editing has become fun for Clipper.

Thank you for your support.

Sincerely,

Donald W. Bullock

APPENDIX C - FORM LETTER AFTER MERGING

THE EFFECTIVENESS OF MULTIPLE LINKED REPRESENTATIONS IN COMMON EDUCATIONAL SOFTWARE

Douglas Frye
Elliot Soloway
Computer Science Department
Yale University
51 Prospect Street
New Haven, CT 06520

ABSTRACT

One of the new ways computers can provide instruction is through multiple linked representations where students learn by being presented with the same information in several different forms. The forms are linked in that changes in one representation will also be reflected in the other. These multiple representations can be successive, follow each other on the computer screen, or simultaneous, reside in different areas on the same screen. The present study evaluated both types of multiple representations as implemented in some typical educational software for personal computers. We tested a word ending game and a simple addition program with 5- and 7-year olds. The results showed that young children who do not already understand these topics can experience difficulty recognizing the links between different representations, especially when the representations were presented successively. These results do not invalidate the technique of multiple linked representations, but they do question the educational effectiveness of the particular implementations used in this software, and argue that the conventions for linking representations must be improved.

Effectiveness of Multiple Linked Representations in Common Educational Software

Computers offer unique advantages and disadvantages as instructional devices. One of their disadvantages is that students must learn to master a piece of computer software at the same time as they are trying to learn the subject domain that the software is presenting. Communication between the student and a human teacher does not, of course, suffer the same difficulty. This disadvantage of computers highlights the importance of interface design for educational software (Frye & Soloway 1987).

Computers offer advantages in that they can present instructional sequences that would be very hard or impossible to produce in any other way. For example, it is known that giving different representations of a problem can be crucial for solving different aspects of the problem (Larkin & Simon, 1987). Computers can provide *yoked* or *multiple linked* representations of a problem (Resnick & Johnson, 1987). For instance, the computer can display an algebraic equation at the same time as it shows a graph of the equation. Changes in the equation will be immediately reflected in the graph. In this way, what is obvious in one representation--the shape of the curve and its crossing points--can help explain the other, thereby giving the student the opportunity to understand the domain.

This paper explores the effectiveness of multiple linked representations as implemented in several types of common educational software for children. The first is an example of what might be called a *successive* multiple linked representation where a representation on one screen is followed by a different representation on the next. Here we studied a program for teaching word endings that presented a written version of a word followed by an animated illustration of the word. We also studied an example of *simultaneous* multiple linked representations. In this case, it was a family of programs for instruction in counting, addition and subtraction that linked numbers on the screen to sets of graphical objects representing those numbers.

The study considered two main questions: 1) Do children make the links between representations as intended by the software? To answer this question we asked set questions of the children during and after they used the software to see if they understood the sequence of events that took place on the screen. 2) Could the multiple linked representations be effective in instructing the children? To evaluate this question, we assessed the children's pre-existing reading and arithmetic skills to see how they would interact with the multiple linked representations offered by the software.

METHOD

Software Tested

The example educational software tested was made up of several commercially available programs produced by CompuTeach, Inc. for use on IBM PC. Although the software came from one company, both the word ending and arithmetic programs were very similar to a variety of other programs now on the educational market. If anything distinguished these particular programs, it was that they incorporated very good quality graphics. The software was designed for children at the late preschool to early primary school level.

Successive Multiple Linked Representations. The wording ending program, Word Pieces, provided instances of successive representations. The program first presented the child with a screen showing a two-letter word ending, like _og, preceded by a blank. The child could then push any letter key on the keyboard and that letter would move across the screen into the blank. If the letter chosen formed a word e.g., dog, then the screen was replaced with a graphical illustration of the word, in this case, a drawing of a large dog (please Figure 1). When appropriate, the drawing incorporated simple animation--the dog's paw moved back and forth. If the child picked a letter that did not form a word, then an irregular shape was drawn on the screen. The shape was meant to resemble the splat of a custard pie hitting the screen. After the splat, the word ending was presented to the child again.

Simultaneous Multiple Linked Representations. The family of arithmetic programs tested made extensive use of simultaneous representations. The addition program is a good example. In this program, as shown in Figure 2, the child was presented with a single-digit addition problem across the top of the screen. Beneath that, two boxes were drawn, one below each number. Each of the boxes was then filled with as many identical drawings of a small object (a car, smiling face, flower, etc.) as in the number above. Finally, "How many?" was written on the bottom of the screen in order to prompt the child to enter an answer to the problem. When the child entered an answer, the numerical and graphical displays changed simultaneously. For a correct answer, the addition problem was replaced by the total number, the boxes were dissolved, and the objects marched off the screen one-by-one while the total decremented to zero. For a wrong answer (after three tries), the addition problem was removed, and each of the objects in both boxes were marked one-by-one with an 'x' at the same time as a number at the top of the screen incremented until the total in the two sets was reached.

Subjects

A total of 20 5-year olds and 20 7-year olds participated in the study. The children were all students at a Cambridge, England primary school. They had all had some experience with computers either at home or at school.

Methodology and Design

The study was conducted in a semi-naturalistic fashion. Each of the students was asked to try out the addition and word endings program on an IBM PC. Order in which the programs were presented was counterbalanced within the two age groups. Each subject attempted 15 addition problems and 10 trials on the word ending program in a session that lasted approximately an hour. The children's responses to the programs were recorded by the computer. The session was also videotaped.

The major data collection technique in the study, besides the students' responses, was a set of probe questions that were asked at specific points during and after the child's experience with each of the programs. The subset of questions relevant to this study were designed to find out which of the representations the children understood and whether they formed the intended links between them. So, on pre-specified trials we asked the children:

For the successive multiple linked representations (word game):

- if they could read the word (or nonword) they had formed;
- when they made a word, if they recognized and could name the picture that followed;
- when they formed a nonword, did they understand the splat or error screen that followed;
- could they ever predict what the picture would be before it appeared.

For the simultaneous multiple linked representations (addition):

- could the child read the addition problem;
- did the child make the connection between the numbers and the objects in the boxes below as measured by:
    - if a box on the screen was covered, but the number above was still visible, could the child say how many objects were in the box:
    - with only one number and the other box visible (which did not contain the appropriate number of objects) does the child know that the two do not go together;
    - after an incorrect response, does the child recognize the error routine of the program putting x's through the objects to find the total as a remediation sequence.

The subjects were also given a number of separate, short tests off the computer. To test skills relevant to the wording ending game, they were examined on their ability to read a list of common three-letter words aloud and they were asked to make a list of as many words as they could with another two-letter word ending slightly different from the one they were given in the Word Pieces game. For skills relevant to addition, they were tested on their ability to count and to do simple addition problems as assessed by a portion of the arithmetic subtest of the WISC.

RESULTS

Word Pieces Game

There were large age differences in the children's proficiency with this game. For example, the 7-year olds produced legitimate words on an average of 7.2 of the 10 trials whereas the 5-year olds only managed to do so on 4.6 trials (t (38) = 4.36, $p < .001$). There were also clear pre-existing differences between the groups in that 18 of the 7-year olds passed the reading pretest compared to only 3 of the 5-year olds (corrected $X^2 = 19.65$, $p < .001$). These differences indicate that the groups represented different ability levels and so can provide a useful comparison for their reactions to the game.

Successive multiple linked representations. A main finding relevant to the effectiveness of the multiple representations was whether the children recognized the picture that was presented after they formed a new word. The results show that the 7-year olds could name 60% of the pictures they saw, but the 5-year olds only recognized 8.3% of them (t (38) = 6.3, $p < .001$). The results for the nonwords were slightly better since 17 of the 20 7-year olds and 10 of the 20 5-year olds (corrected $X^2 = 4.1$, $p < .05$) understood that the splat meant they had formed an incorrect word. The children who understood what the splat meant came to that understanding relatively quickly. On average, it took just 1 trial for the 7-year olds and 2 trials for the 5-year olds. At the end of the session we asked explicitly if the children could explain the relationship between what they entered on the screen and the pictures that followed. There were 16 7-years olds and 3 5-

year olds (corrected $X^2 = 14.4, p = .001$) who could.

Addition Game

Group differences were also apparent on this task. Of the 15 simple addition problems, the 7-year olds answered a mean of 13.4 correctly as compared to 5.65 for the 5-year olds (t (38) = 5.88, $p < .001$). The groups also differed in their ability to count to 20 without error; all of the 7-year olds could versus 9 of the 5-year olds (corrected $X^2 = 12.54, p < .001$). Furthermore, the 7-year olds outscored the 5-year olds on the addition problems from the arithmetic subtest of the WISC (t (38) = 4.46, $p < .001$).

Simultaneous multiple linked representations. The symbolic representations in the addition game caused the same difficulties for the younger children as they did in the word ending game. Here, 14 of the 20 7-year olds could read both the numbers and plus sign in the addition problem while only 2 of the 5-year olds could (corrected $X^2 = 12.6, p < .001$). Both groups were poor at determining the meaning of the error sequence that accompanied a wrong answer. Only 7 of the 7-year olds and 4 of the 5-year olds were able to interpret what it meant. There was some evidence that a portion of the children understood the relationship between the numbers and the boxes of corresponding objects. When a box was covered, 15 of the 7-year olds and 10 of the 5-year olds could say how many objects were there if the number above was visible. When the the wrong number and box were paired, 9 of the 7-year olds and 6 of the 5-years correctly objected. A specific question after the game about the relationship between the numbers and the objects below brought correct answers from 14 of the 7-year olds and 9 of the 5-year olds.

DISCUSSION

The results of the study question the effectiveness of the successive multiple linked representations in the word ending game. This program sought to teach new vocabulary and beginning reading skills. It had the child choose any letter to be added to a word ending which was then followed by a screen showing a drawing representing the word's meaning or, in the case of a nonword, an error screen. This arrangement was not effective for the 5-year olds. They recognized fewer than 10% of the pictures. Only half of them realized that the error screen was signalling an error. Just 3 out of the 20 could indicate the relationship between forming a word or nonword and the screens that followed. The results were better for the 7-year olds. They recognized 60% of the pictures. All except for 3 or 4 of them understood the error sequence and knew the relationship between their response and the screens that followed. It must be noted, however, that the 7-year olds were much better at forming words in the game and many showed evidence of already being able to read. As such, it could have been that their understanding of the words they formed helped them interpret the pictures rather than vice versa.

The results for the simultaneous multiple linked representations in the addition game were more varied. This program attempted to teach addition by showing sets of objects linked to simple addition problems. When the child entered an answer, it too was presented along with a set of objects. Those objects enacted different routines depending on whether the answer was right or wrong. In this game, although only 2 of the 20 5-year olds could read the addition problem, about half of them showed some evidence of knowing the link between the numbers and objects on the screen. As before, the results were different for the 7-year olds. Nearly three-quarters of them could read the addition problem as well giving evidence they understood the link between the numbers and objects. The error sequence in this game, which also depended on simultaneous multiple linked representations, was not a success with either group. Only 7 of the 7-year olds and 4 of the 5-year olds could identify what it was.

On the whole, neither form of multiple linked representation seemed highly advantageous as an instructional aid, although the simultaneous representations may have had more of an effect. Many of the 7-year olds could cope with the linked representations, but they may have been too skilled in reading and addition to provide a stringent test. In fact, in the word ending game, it was not unusual that instead of learning the word through the picture that followed, the older children relied on their understanding of the word to interpret the picture. The 5-year olds were at a level where they could have learned from the software. The word game seemed to fail in this respect because the younger children recognized very few of the pictures and they showed little evidence of understanding the link between the successive representations. In contrast, there were indications that the simultaneous representations in the addition program may have made an impression. Although very few of the 5-year olds could read the addition problems, nearly half of them linked the number and object representations on the screen.

It must be noted that this study only tested two particular implementations of multiple linked representations. Other implementations may be more effective than these were. It is also plain that the comparison of the successive and simultaneous representations was confounded by other differences. The most obvious being that one was conveying instruction in reading and the other in arithmetic. It may be that the successive linked representations would have been more effective if they too had been in a mathematical domain. Nonetheless, these implementations stand as typical examples of how linked representations are now being employed in popular educational software. Furthermore, the results have value for the problems that they reveal about implementing multiple representations.

The pictures in the word game provide a good illustration of some of these problems. Although all of the children recognized the drawing of the dog when it was shown, they were not adept at deciphering the picture of a cog or even the one of a log. Since of all these words are simple nouns, they are probably the easiest parts of speech to explain with a picture. Given the difficulties even these words caused, the limits of this technique for language learning are apparent.

On a more general level, the results indicate that the connections between representations that are clear to adults--i.e., experts in the domains of reading and arithmetic--may not be obvious to students just learning these skills. In an unfamiliar domain, why should one perceive links between a screen that follows another or between different regions of the same screen? Young children most likely connect pictures with text in books because adults emphasize the relation between the two very early on. The child may not necessarily receive similar adult assistance with the computer, so other approaches may have to be pursued. In the word ending game, it might have been useful to show several easily identifiable words and pictures first, thereby giving the child the chance to infer the link between successive representations. The simultaneous linked representations might have been improved if the game had been more interactive. Specifically, more opportunity could have been given for the child to manipulate one representation in order to see the allied effect in the other.

Multiple linked representations make available a new instructional technique that is really only feasible by virtue of the computer. This technique seems to have the potential for teaching new skills and knowledge rather than just giving students practice with what they already know. The technique is becoming popular. For example, hypertext (as exemplified by Apple's recent HyperCard program) is in essence an entire system for linking successive and simultaneous multiple representations. The results of the present study do not argue against multiple linked representations. They do argue that for the educational promise of this technique to be fulfilled, we must develop representations and conventions for linking them that students will readily understand.

REFERENCES

Frye, D. and Soloway, E. (1987). Interface design: A neglected topic in educational software. In J. Carroll & P. Tanner (Eds.), *Human factors in computing systems and graphics interface*. Amsterdam: North-Holland.

Larkin, J. and Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65-99.

Resnick, L. and Johnson, A. (1987). *Intelligent machines for intelligent people: Cognitive theory and the future of computer-assisted learning*. Unpublished manuscript, Learning Research and Development Center, University of Pittsburgh.

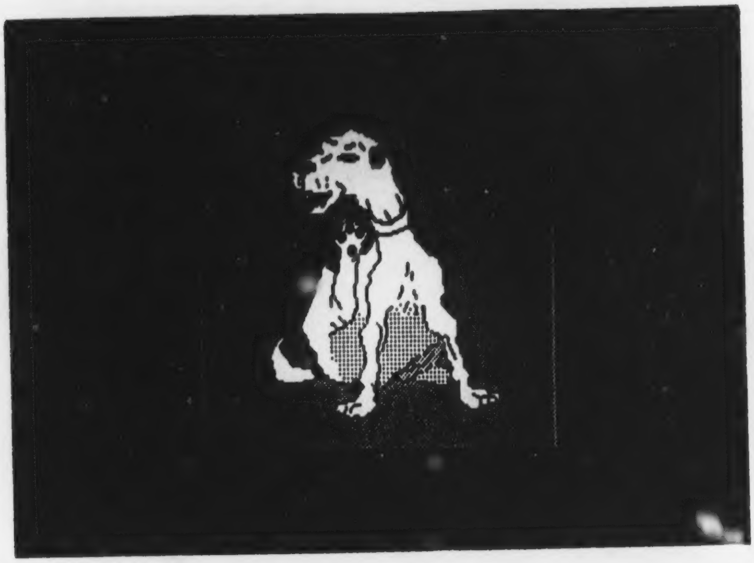


Figure 1. An example picture from the wording game.

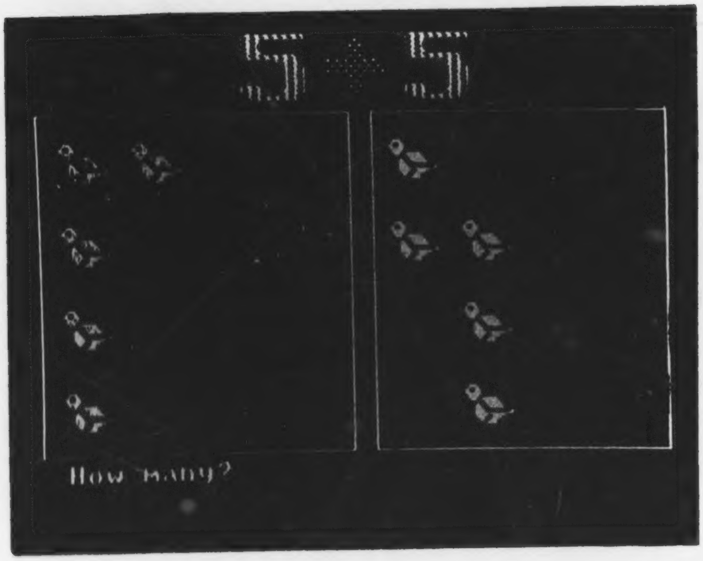


Figure 2. An arithmetic problem as presented by the addition program.

SOFTWARE TOOLS FOR A VIRTUAL CLASSROOM

Ellen S. Schreihofer
John L. Foster
B.J. Gleason
Heidi E. Harting
Starr Roxanne Hiltz
Computerized Conferencing and Communications Center
New Jersey Institute of Technology
Newark, New Jersey 07102
(201-596-5769).

[The material in this paper was adapted from information in the following publications: Branching Activities in Conferences by Starr Roxanne Hiltz ©1987; Instructional Management Support Tools on EIES by BJ Gleason ©1987; Final Design Specifications for Personal TEIES by John L. Foster ©1987; and The Virtual Classroom: Building the Foundations by Starr Roxanne Hiltz ©1986.]

EIES, TEIES, Personal TEIES and Virtual Classroom are trademarks of New Jersey Institute of Technology.

ABSTRACT

During the past decade, the Computerized Conferencing and Communications Center at the New Jersey Institute of Technology has been involved with computers in education. During the past two years, a three year project was begun, funded by the Annenberg/Corporation for Public Broadcasting Project, which is concerned with the development and assessment of new types of software capabilities designed to support university level courses. A 'Virtual Classroom' or 'university without walls' is being created within a computerized conferencing system. The goal of the project is to develop and evaluate this major new means of educational delivery for college-level education.

During the first year of the project, students in twelve courses at three universities completed part or all of their coursework 'online.'

Software is being developed to support the 'Virtual Classroom.' TEIES (Tailorable Electronic Information Exchange System), a computer mediated communication system, is under development at the Computerized Conferencing and Communications Center under a joint study contract with IBM. TEIES has several objectives, including servicing frequent users such as students using TEIES to augment all of their courses or enroll in remote education courses delivered entirely on TEIES.

Personal TEIES is an IBM PC-based graphics text editor and display system which is also being developed to interface with existing messaging and conferencing systems, data bases or file systems. Development of a conferencing-oriented color graphics capability was undertaken so that equations, diagrams and other non-text material may be transmitted online, to augment the Virtual Classroom.

Other software development on the currently used computerized conferencing system, the Electronic Information Exchange System (EIES), was created that would facilitate and build the Virtual Classroom. Instructional tools such as electronic gr adebooks, quiz and test-taking capabilities, automatic graders, and 'branches' (automated activities such as topic branch, document read and material selection) for class conferences were developed and implemented in the online courses.

OBJECTIVES OF THE VIRTUAL CLASSROOM PROJECT

The goal of the Virtual Classroom project is to develop and evaluate a new means of educational delivery of college-level education. A Virtual Classroom is a teaching and learning space located within a computer-mediated communication system. In order to be considered a "Virtual Classroom," the system must support all or most of the types of communication and learning activities available in traditional (physical) classrooms. This includes class conferences where teachers may "lecture" and students may discuss course material seminar-style (group discussion); a messaging system for private communication between student and teacher and among students (a communication structure similar to "office hours" or private discussions); and special software for activities such as tests and graphics creation and display. Software was created to support the ability to administer, collect and grade tests or assignments; and the ability to divide a large class into smaller working or peer groups for collaborative assignments. There is also the equivalent of a "blackboard" where diagrams or equations may be posted.

A major difference between a Virtual Classroom and the traditional classroom is that in the traditional classroom, most interaction takes place by speaking and listening (supplemented by handouts, and the blackboard.) In the Virtual Classroom, communication takes place almost entirely through interaction with a computer terminal. The Virtual Classroom is supplemented by occasional face-to-face meetings and telephone calls, and uses printed material such as textbooks. In addition, because the Virtual Classroom is located within a computer-mediated communication system, interaction in the classroom is asynchronous, with the computer storing and managing communication for each participant.

The Virtual Classroom is not intended to be a "second best" simulation of a traditional classroom. On the contrary, a collaborative learning environment that is computer-mediated can support types of activities that are impossible or difficult to conduct in face-to-face environments. Both traditional and Virtual Classrooms have their own strengths and weaknesses. The effectiveness of a Virtual Classroom is dependent on the teacher conducting the course in a manner which fits the characteristics of this medium, the nature of the course material, and the characteristics of the students. A major factor involves adequate access to the necessary equipment (personal computer and modems), so students can easily participate in their class. The basic premise of the Virtual Classroom project is that given the proper software tools and depending on these contingencies, the Virtual Classroom may be a more effective mode of delivery for post-secondary education than the traditional (physical) classroom.

The objective of the "Virtual Classroom" is not only to replicate the forms and modes of interaction and activities available in the traditional classroom. In addition, we are attempting to use the power and characteristics of the computer to produce interactive group learning activities which improve on the traditional classroom in three major ways:

Convenience of access to educational opportunities- Each student can participate at a time and place which is most convenient. He or she enters the Virtual Classroom by dialing its number and connecting via a home or office microcomputer or computer terminal. The classroom is brought to the student, rather than the student travelling to the class.

More active participation- In this medium, the student may be encouraged to participate more actively than in a traditional classroom. A faculty member can use the computer to force individual thinking and responses and group discussion and processing of ideas and concepts; the student can be prevented from being merely a passive listener or watcher. A goal of the Virtual Classroom is to facilitate and combine "active learning" and "peer-group or collaborative learning" (Collier, 1980; Bouton and Garth, 1983), which may provide more effective learning.

Self-pacing- Because the material is stored and mediated through a computer system, each student can proceed through the material at an optimal pace for him or her.

The ultimate objective is to create a new medium of educational delivery that will provide a higher quality as well as more conveniently obtained educational opportunity, for at least some types of students and course materials. It is expected that the products developed through this project will provide a superior educational experience for certain types of students and selected types of course materials.

The intended long-term outcome of these activities is to make available Virtual Classroom software that will allow any university to use a computer to offer courses in this new mode. Development efforts for the project will include rewriting the software tools designed and implemented on EIES1 in the "C" language and implement these on TEIES (Tailorable Electronic Information Exchange System), which will initially operate in the IBM VM environment. The offering of courses via the new medium may vary from a single course offered via a version of TEIES that runs on a super-micro (such as the IBM-AT) with only a few ports, to an entire "electronic university," running on a dedicated mainframe that could accommodate up to 10,000 students.

The project takes advantage of the spread of computers into homes and offices to make them the means of delivery of higher education, especially for adult learners; a means that is potentially less costly and more effective than any current alternatives. For this project, software development and evaluation are equally important goals.

SOFTWARE DEVELOPED TO SUPPORT THE VIRTUAL CLASSROOM

BRANCH

To facilitate interaction in the Virtual Classroom, several software programs were developed and incorporated into the existing computerized conferencing system (EIES). All online classes transpire in a private class conference. The class conference is where teachers may lecture to the entire class and students may respond with questions, answers, comments, or homework, which is also presented to the entire class.

Conferences are designed to be linear, starting with conference comment number one, and adding one comment after another in a traditional chronological order. Conference comments may be entered by either the students or the teacher, and the material contained in a comment may be an "e-lecture" (electronic lecture), assignment, question, student feedback, or any information presented in the class. In an effort to maintain control over the discussion in a conference, "activities" were developed.

BRANCH is the name of the application used to describe activities which are attached to comments in a conference. The main conference comments form a "trunk;" and the "branches" attach to one of the main conference comments. Branches may be created as "sequenced units", groups of branches that must be completed in the order specified by the author; or as "single/free access" branches, which may be done in any order desired.

Consider the conference as a tree, with the main discussion progressing in a linear fashion up the "trunk" of the tree. The need to have "branching" discussions of particular topics, without affecting the main conference led to the development of "BRANCH activities." BRANCH activities organize topics and assignments in logical groupings. All of the responses or activities related to that branch are gathered together, rather than being scattered

There are three types of branches currently available, RESPONSE BRANCH, READ BRANCH and SELECTION BRANCH. The most frequently used type of branch activity is RESPONSE branch. The purpose of a RESPONSE branch is to contain the discussion of a single topic to one place in a conference.

All conference comments are presented in the main conference (trunk) to which a discussion may follow. RESPONSE branches link the responding comments to the main conference comment. Each branch response maintains the same conference comment number with the suffix ".#" (the "#" standing for the order in which the responses were entered). The author of the initial main conference comment may specify if multiple answers per person may be entered in the branch. Other conditions may be specified by the teacher, such as the stipulation that a student must enter a response prior to viewing other responses in the branch. This is important to ensure each student thinks and answers independently, without being influenced by the responses entered by others. Likewise, the author of the main response branch may specify that participants are permitted to view the responses of others prior to adding their own response.

The author of the branch may also regulate how the response entry must be signed (anonymous signature, pen name, or full name). Use of a "pen name" may be useful in allowing students to feel less inhibited in communicating about personal feelings, or when role playing and assuming the character of another personality. If the teacher did not select the option of allowing pen name entries in the conference, the traditional signature (students'full name) would be automatically signed to the comment.

A second type of branch activity is the READ branch. The purpose of a READ branch is to organize large documents into smaller sections and/or link smaller articles into a logical arrangement. This allows essays or lecture type material to be divided into sections. Each section has a title, and may be read by selecting the particular section from the table of contents. When participating in a READ branch, the student has the option to read the entire entry, or specific parts of the "read" branch.

The final branch activity currently available is the SELECTION branch. The SELECTION branch was designed to structure and regulate a selection process. A topic and a list of items to be selected is contained in a main conference comment. The author of the branch determines how many selections each participant may choose. Once a user has chosen an item, his/her name is posted with the item and it is no longer available for selection by others. An application for this type of activity would be for students to select assignments/topics, such as term papers. Once the topic has been selected, it is automatically removed from the available list and is posted next to the students name.

The main BRANCH CHOICE menu provides the many options available when doing branch activities. Below is an example and explanation of how the main branch options appear on the existing computerized conferencing system (EIES).

```
                BRANCH CHOICE MENU

     Choose From:

          Get Branch                  (1)
          Display Branch              (2)
          Review Branch               (3)
          Do Branch                   (4)
          Modify/Delete item          (5)
          Author/Create Branch        (6)
          Set Interaction Mode        (7)
          Monitor                     (8)
          Create/Modify Unit          (9)

     BRANCH CHOICE?
```

A question mark (?) entered at BRANCH CHOICE will provide the following short explanation of Branch Choices:

1 Gets the root comment for a branch item, header plus text.
2 Displays the header for the root comment of a branch.
3 Reviews all branch items and the users status on completing each one.
4 "Do" branch enables the participant to respond to a RESPONSE branch, read a READ branch, etc.
5 Allows user to modify or delete a response or branch which he/she has authored.
6 Allows user to create a branch IF the moderator of the conference assigned that privilege.
7 Allows switching to a "batch" mode whereby items are delivered automatically.
8 Allows the conference moderator to change the way existing branches work, and to review activity in branches.
9 Allows moderator to organize branches into sequenced units.

Future development of BRANCH activities will continue on TEIES (Tailorable Electronic Information Exchange System), a computer mediated communication system under development at NJIT. Branch will be incorporated into all conferences as a standard feature on TEIES.

INSTRUCTIONAL MANAGEMENT SUPPORT TOOLS

Continuing with the development effort to facilitate online classes, several instructional management tools were created. These consisted of software related to the many aspects of creating, presenting, and grading quizzes/tests; electronic gradebooks; "handing in" assignments; and accessing routines to compile and execute PASCAL and FORTRAN computer programs (developed for online classes which involved computer programming). Quizzes were designed to accommodate four types of responses:

EXACT - Allows quiz taker to input one line answers.

YES/NO - Similar to "true/false" type questions. Requires students to answer either "yes" or "no".

NUMBER - Permits essay-type questions. Allows students to compose a specified number of lines of response.

MINMAX - Multiple choice questions, where selection is made from a range of numbers.

For security of the quiz information, only the instructor may access the quiz preparation facility. The student may not access the quiz until the teacher had specified that the quiz may be presented to the entire class. Features of the online quiz include a time limit restriction which notifies the instructor if a student had extended past the alloted amount of time; and "one-time" student access to the quiz, which attempts to minimize cheating. This restricts the student from signing on, getting the question and then signing off to get the answer. In addition, if a student tries to log off while taking a quiz, the instructor is notified of this and the answer as they have prepared it is automatically submitted to the teacher.

Related to the quiz activity was development of a simple way to obtain student results of a quiz. The teacher may get a summary indicating the students who have completed the quiz. Automatic grading of the information is also available. The grades are entered into the instructor's electronic gradebook, and the teacher has the option to automatically notify the student of the grade received. Only "YES/NO" and "MINMAX" response items may be automatically graded. Other types of quizzes need to be graded by the instructor.

Students may obtain a summary of the assignments completed and the grade received for each assignment at any time. This enables the student to maintain a clear idea of their grade status in the class.

To facilitate "handing in" an assignment by the student and reviewing the assignment by the teacher a HANDIN support tool was created. HANDIN is a way for students in the Virtual Classroom to privately compose their material and then easily send completed assignments to their instructor. Routines were also developed for online computer programming classes to submit source code and data files for PASCAL and FORTRAN programs to a compiler that compiled and executed the program and returned the results to the student.

As student and instructor needs are continually assessed, other instructional management tools may be developed to help create the Virtual Classroom.

PERSONAL TEIES

One of the most fundamental software development efforts undertaken in the Virtual Classroom project is the creation of Personal TEIES. Personal TEIES is designed to be a front-end software package to the mainframe EIES and TEIES conferencing systems. Personal TEIES has several components, including a communications program, a full screen text and graphics editor, and communications management functions.

Internally, Personal TEIES is written in the "C" language using the Graphical Kernal System (GKS) standard. GKS provides a standard set of display functions, controls and attributes. The North American Presentation Level Protocol Syntax (NAPLPS) is the standard used for data communication. It provides the capability to send graphical information from machine to machine and to store graphics in the data base.

Personal TEIES was developed for students who use basic IBM-PC compatible machines. Since this was the target audience, the initial version of Personal TEIES was developed to operate on any IBM-PC compatible personal computer with atleast 256k, two disk drives and graphics display capabilities. Future extensions of the Personal TEIES software would include portability to higher end work stations such as SUN, APOLLO, and MAC+. Future development will improve the functions of the graphics and information management and move the system to higher level workstation type computers.

SCRATCHPAD

The main feature of Personal TEIES is the Scratchpad, an integrated full-screen text and graphics editor. Simple graphic composition tools are used, such as line, box, circle, arc, point, text and free-hand draw. Multi-paged communications can be created and edited quickly and easily. With only a few functions available the user interface is easy to learn. The purpose is to stimulate communications between individuals, not to concentrate on high quality publication style documents. Composition editing functions include copy, move, cut, paste, rotate and size. Any of these editing functions may be applied to graphics objects, text, and icon symbols (described below). A set of options allow color changes, line and fill pattern styles, and character font changes.

One unique feature of the Scratchpad is the Icon Generator. This feature allows the user to create and edit libraries or sets of icons. Icons are symbols that are pictorial representations of higher level concepts. An icon is a pixel image of a graphical object. The icons used in Personal TEIES may be pictures that can stand alone or may have textual information included. Pictures are displays that contain symbols, icons and graphical elements created within the system through the use of graphics primitives. Pictures may be a simple page of ordinary text, or a complex free form drawing.

Typically icons are used to represent interface functions, however, in TEIES the application is more advanced. Users may define their own icons and define rules governing the use of an icon, and then distribute sets of icons to other users. Example uses of icons are for Electrical Engineering circuit diagrams, Architecture layouts, Computer Science flow charts and data flow diagrams, and Mathematical symbols for equations.

The Icon creation or editing process includes: defining the physical appearance of the symbol; a textual description of the symbol any user can see through a pop-up window request; the rules for using the icon in a composition; and its keyboard position for recall. Each icon is mapped to an alternate keyboard key for quick access during the composition process. If the corresponding keyboard key is not known for a symbol, a help window shows the alternate keyboard layout.

Symbols consist of not only linguistic alpha-numeric characters, but also the specialized characters used in mathematics and other technical fields such as chemistry, engineering, architecture, statistics, music, computer science, etc. Symbols may be used to construct sentences, equations and in general must be used in conjunction with other symbols to convey complete information. Since Personal TEIES is designed to be a software front end to a conferencing system, asynchronous communication may then exist between a personal computer and the mainframe; from personal computer to personal computer; or communication may exist between several personal computers connected via Local Area Networks (LAN).

In addition to individual composition of graphics and text, Personal TEIES encourages "group graphics." There is the application of a "blackboard", which enables group discussion with a drawing and a collaborative effort for document composition and editing, proposal writing, and distribution of documents. A graphics/text item may be created by one person, but revised and expanded upon by others.

Another application of group graphics would be when the teacher presented an assignment with a defined solution. Personal TEIES enables the students to place symbols in a diagram, and the software compares the results against the defined solution.

Personal TEIES facilitates the distribution of icon sets and "boiler plate" documents to be utilized, revised and re-submitted in a group effort. The group graphics capability also enables features of the Virtual Classroom, such as BRANCH activities and quizzes to include graphical illustrations and composition.

Future development of Personal TEIES would primarily involve further investigation of the group collaborative process and the creation of sophisticated communication with the mainframe.

SUMMARY

Prototypes of software to support collaborative learning activities and instructional management in a "Virtual Classroom" were initially prototyped within EIES and used in a variety of courses. These are currently being implemented in a new computer-mediated communication system (TEIES), which is available to universities for lease or purchase.

On a comparative basis at both the New Jersey Institute of Technology and Upsala College, the Virtual Classroom software has been used to totally conduct the following courses online and compare them to traditional face-to-face sections:

Upsala College: Introduction to Sociology and Computer-Assisted Statistics.

New Jersey Institute of Technology: Introduction to Computer Science and Statistics for Technology.

Other courses where the Virtual Classroom software has been used to partially replace face-to-face classroom meetings or to augment those class sessions include:

New Jersey Institute of Technology: Computers and Society; Evaluation of Information Systems (graduate level); Management of Computer Systems; Design of Interactive Systems (graduate level) and Management Practices.

Upsala College: Freshman Writing Seminar; Anthropology-North American Indians; French Correspondence Course; Data Analysis and Computers and Society.

Other schools offering totally online courses using the Virtual Classroom facilities include the New School in New York, through Connected-Education, Inc. Courses in the Media Studies program include Ethics in the Technological World; Tele-Law; Applications in Telecommunications; and Computer Conferencing in Business and Education.

The results of the above evaluations are available in various reports and published papers (Hiltz 1986, 1987), noted in the Reference section.

REFERENCES AND INFORMATION

Bouton, Clark & Garth, Russell Y. (1983), Learning in Groups. *New Directions in Teaching and Learning*, no. 14. San Francisco: Jossey-Bass.

Collier, K.G. (1980), Peer-group Learning in Higher Education: The Development of Higher Order Skills. *Studies in Higher Education*, 5,1, 55-62.

Foster, John, (1987), Final Design Specification for Personal TEIES: Text and Graphics Composition System and Personal Communications Manager Technical Report 86-15.2, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

Gleason, B.J., (1987), Instructional Management Tools on EIES, Technical Report 87-16, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

Harting, Heidi, (1986) User Manual for Personal TEIES, Technical Report 86-4, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

Hiltz, Starr Roxanne, (1986), Branching Capabilities in Conferences: A Manual and Functional Specifications, Technical Report 86-1, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

_____, (1986), "The Virtual Classroom: Using Computer-Mediated Communication for University Teaching," *Journal of Communication*, 36:2, 95-104.

_____, Johnson, Kenneth, and Turoff, Murray, (1986), "Experiments in Group Decision Making: Communication Process and Outcome in Face-to-Face

Versus Computerized Conferences," *Human Communication Research*, 13, 2, 225-252.

_____, (1986), The Virtual Classroom: Building the Foundations, Research Report 24, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

_____, (1987), Learning in a Virtual Classroom: A Virtual Classroom on EIES: Final Evaluation Report, Research Report 25 volume 1, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

_____, (1987), Teaching in a Virtual Classroom: A Virtual Classroom on EIES: Final Evaluation Report, Research Report 26 volume 2, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, NJ.

Turoff, Murray, Hiltz, Starr Roxanne, and Foster, John, (1987) TEIES-(Tailorable Electronic Information Exchange System): Overview of Interface Design, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark,

_____, Hiltz, Starr Roxanne, (1985), "Remote Learning: Technologies and Opportunities," invited paper for the World Conference on Continuing Engineering Education, May 1986, Lake Buena Vista, Florida.

Figure 1. Response Branch

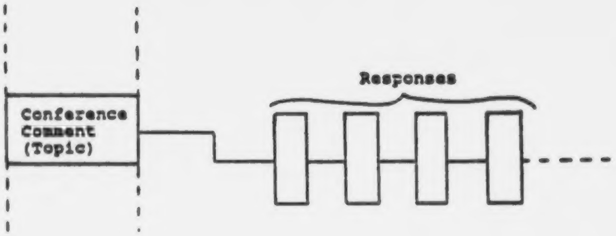

Conference
Comment
(Topic)
Responses

Figure 2. Read Branch

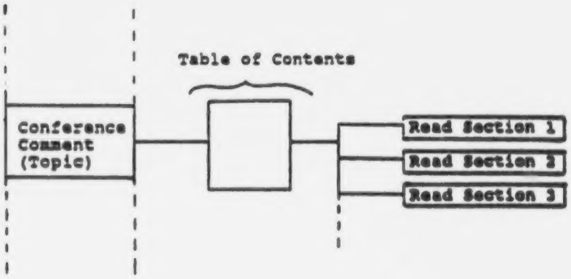


Table of Contents
Conference
Comment
(Topic)
Read Section 1
Read Section 2
Read Section 3

Figure 3. Selection Branch

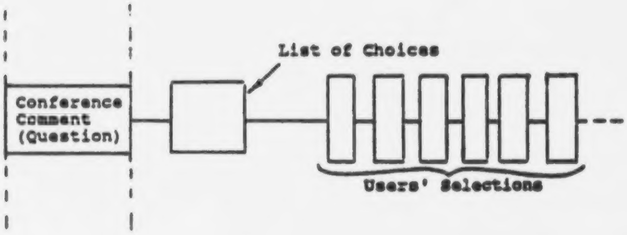

List of Choices
Conference
Comment
(Question)
Users' Selections

Figure 4. PTEIES create ability

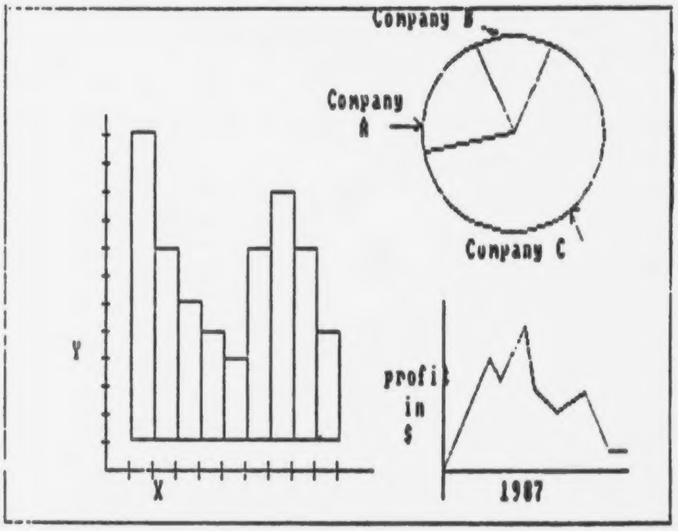
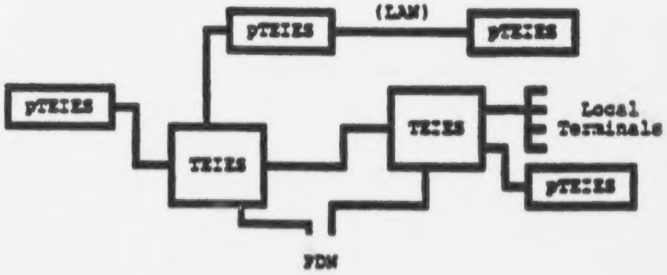


Figure 5. TEIES Network Topology


pTEIES
(LAN)
pTEIES
pTEIES
TEIES
TEIES
Local Terminals
pTEIES
PDN

COMPUTER SCIENCE

Andrzej Kornecki
Ed Lewis, Richard L. Baskerville
Rodney O. Rogers
Gary W. Tubb

ABSTRACT: Microprocessor Laboratory in a Software-Oriented Computer Science College Program

Dr. Andrzej Kornecki, Associate Professor, Computer Science Department, Embry-Riddle Aeronautical University, Daytona Beach, FL 32014

Most of the undergraduate computer science programs, particularly in liberal art colleges, are software oriented. One reason for that is the development of computer science as a theoretical discipline dealing with algorithms, databases, compiler writing, and so forth. The other reason is that due to the expenses, only limited courses are offered on hardware issues, usually without adequate laboratory support. One can also point out the lack of qualified individuals to teach these courses. The microcomputer revolution has reached the point the everyone seems to be a computer expert. However, it is amazing to realize how little the average computer professional understands how microcomputer systems work. It is imperative for future computer science graduates to develop understanding and gain more detailed knowledge of the principles of internal microcomputer operation. Microprocessor architecture, instruction list, signal levels, interrupts, memory and input/output interface are essential topics. The details of a disk operating system, interfacing disk drives, displays, printers, and modems are necessary to provide a complete picture of a microcomputer system.

Following these considerations, Embry-Riddle Aeronautical University introduced a microprocessor laboratory for computer science majors. In the lab we have 10 two-person student stations equipped with the Intel 8086-based microprocessor development boards and external input/output boards with switchs/LEDs. There are also an IBM PC running macroassembler, a debugger, and *The Visible Computer* (software stimulating internal operations of the Intel microprocessor). A PROM programmer with developed software permits students to upload and download files and facilitate a simple system development. A logic analyzer is used for presenting concepts of software debugging and hardware troubleshooting. Assorted tools (multimeters, pulsers, probes, signal gererators) and components (integrated circuits, passive elements, connectors) are used for the development of experiments.

Introducing such a relatively inexpensive hardware-oriented lab in 1985-86 improved the situation significantly. The laboratory serves computer science students both as a place for instructor demonstrations and for individual experiments in a wide range of computer science courses conforming AMC78 curriculum. Such courses as Digital Logic, Microprocessors, Computer Architecture, Interfacing, and Real Time Systems could be taught in a better and more explanatory fashion. The lab gives students hard-to-get, hands-on experience. At Embry-Riddle we have had experience with the delivery of high-quality, practical experiences with moderate investments in equipment.

ABSTRACT: "Emmeylou," an Intel 8085 Teaching Emulator, as Used in Computer Architecture Instruction

Ed Lewis, Director of Computer Services, McCallie School, Chattanooga, Tennessee Richard L. Baskerville, Professor of Computer Science, University of Tennessee at Chattanooga

In teaching a beginning course in computer architecture, it is often difficult for students to visualize or conceptualize the internal workings of a central processing unit. The movement of data and other information to and from registers, accumulators, busses and arithmetic logic units must be taken on faith and often can be confusing. Project "Emmeylou" was designed to graphically present the internal workings of a simple processor, the Intel 8085, in such a way that the student can halt the processing at any point and view the contents of all the processor components.

The emulator is written in Pascal with few deviations from standard syntax and is highly portable. The source code identifiers describe the function they serve in order that a student may examine the code instructively. The 64K memory accessible to the 8085 is emulated in an array. The 256 op codes and the actions they direct are held either in an array or records or a file, depending on the host processor. Programs for the emulator may be created in decimal, hexadecimal, or binary and the output may be viewed in any of the three modes. Several memory speeds and clock speeds may be set to induce wait states in some instructions.

262

"Emmeylou" is used in the beginning computer architecture course at the University of Tennessee at Chattanooga. One of the goals of this course is to have the student understand architectural behavior at the register transfer level. In the past, the main instrument for this teaching had been the processor "bread board" with the single register hexadecimal display. "Emmeylou" is much less time consuming to learn with a much friendlier interface. The capacity to display all of the registers as well as the flags and busses gives a much more complete learning experience. Students are required to write and debug several programs.

This emulator is now in use in three university environments. The source and files are available through the M.U.S.I.C. users group or the University of Tennessee at Chattanooga. The system is supported by U.T.C. and the author.

ABSTRACT: Marrying Theory and Practice: Teaching Formal Language Concepts to Small College Computer Science Majors

Dr. Rodney O. Rogers, Department of Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL

ACM Curriculum '78 specifies two upper level courses on formal language concepts: CS 15, Theory of Programming Languages; and CS 16, Automata, Computability, and Formal Languages. CS 16, theoretical approach, focuses on the Chomsky hierarchy of grammars and the languages these grammars generate. The automata which recognize these languages are studied in detail, as are the properties of regular sets and context free and context sensitive languages. CS 15, by contrast, is usually more practically oriented. The course treats first four phases of compilation (scanning, parsing, intermediate code generation, and optimization). Top down and bottom up parsing methods for context free languages are examined, as are methods for resolving context sensitive issues. Typically the student is required to write the front end of a compiler for a small language. Often the parser for this project combines recursive descent and operator precendence parsing methods.

CS 15 and CS 16 are among the most important and challenging courses in the undergraduate computer science curriculum. They account for a substantial amount of the material examined on the GRE computer science subject exam. Nevertheless, probably because they are admittedly abstruse, they are perhaps too frequently omitted from small college computer science curricula. At Embry-Riddle Aeronautical University we have developed a course which combines the basic parts of both CS 15 and CS 16 but omits their more arcane aspects. The approach moves from the practical to the theoretical. The student is first introduced to scanners and parsers. He writes a recursive descent interpreter for a language with just one variable type (integer), learning along the way how to emit 3 address code for assignment statements. Of course a symbol table is maintained. Bottom up parsing methods are merely described. The LL(1) top down table driven method is treated likewise. Subsequently, the student is introduced to language theory concepts relevant to the programming project he has just completed. This involves teaching just the first "half" of CS 16: regular expressions and regular sets; the context free languages; and automata relevant to these classes of languages. All the important basic theoretical results are presented, but proofs are not labored. Linear bounded automata, Turing machines, uncomputability, and the halting problem are merely mentioned. The approach is still quite challenging for some students, but it has proven to be a useful compromise in the curriculum of one smaller technically oriented college where Curriculum '78 is deemed an ideal if not an entirely practical goal.

ABSTRACT: A Pedagogy for Prolog Instruction using Turbo Prolog graphics

Gary W. Tubb, Ph.D., Instructional Computing Department, University of South Florida, Tampa, Florida

A pedagogy for instructing the programming of Prolog is presented in this paper. The focus of the instructional method is graphics. The system of learning is based upon the process often used in LOGO to acquire LISP concepts. Fundamental programming concepts of block structure (procedure, local variable) and control structures (recursion) of LISP are introduced in LOGO within a limited data structure - graphics. The proposed Prolog instructional model is couched within the AI area of two-dimensional pattern recognition and capitalizes upon the graphics capabilities of Turbo Prolog. The paper addresses the concepts of facts, goals, variables, backtracking, and rules introduced by using Turbo Prolog graphics. The instructional model follows Sowa's (1984) conceptual structures to a degree. Pietrzydowski's PROGRAPH and LOGRAPH are discussed as possible alternatives to the proposed instructional model.

KEYBOARDING AND FUNDING FOR TEACHING EDUCATION: ELEMENTARY

Gary A. Carnow
Mary Jane Ford, Virginia Poe
Ruth Kerr
Hilary Cowan

ABSTRACT: Elementary Keyboarding: FrEdTyper and Other Alternatives

Gary A. Carnow, Ed.D., Director, California Model Technology Schools Project at Alhambra, Coordinator of Instructional Technology Education, K-12, Alhambra School District, Alhambra, CA

Alhambra School District, home to a multi-million dollar California Model Technology School Project, has devised a number of programs to support keyboarding in the elementary classroom. The 1987-88 year featured a focus on keyboarding as a gateway to an effective computer education program. A study was conducted in district schools to determine the effectiveness of two methods. Students involved in the study received keyboarding instruction either on FrEdTyper or on keyboard alternatives. This presentation describes the study.

Some students used FrEdTyper, a keyboarding program built inside of the "prompted writing" boxes of FrEdWriter. FrEdTyper is a no-cost alternative to keyboarding programs. The software was developed as part of a California Developmental Grant Project based in the Alhambra School District. The software is in the public domain. This program features keyboarding lessons taught in conjunction with word processing skills.

Other students used portable keyboards with an LED read-out, allowing a classroom teacher, for the cost of one computer, to put a keyboarding tool in the hands of every child.

The results of this study will help to clarify:

- *What keyboarding skills are gained by fourth grade students who receive thirty minutes per day of keyboard training for six weeks?*
- *What keyboarding gains are made on computers as compared to the alternatives?*
- *What are the cost-effectiveness benefits to the methods employed?*

The data from this project provides timely information to districts wanting to:

- *implement keyboarding in an elementary school;*
- *utilize the talents of high school business educators as trainer-of-trainers;*
- *embark on meaningful technology staff development;*
- *develop a language arts program with a keyboarding/word processing emphasis*

ABSTRACT: The Acquisition of Microcomputer Keyboarding Skills: Dvorak vs. Qwerty

Dr. Mary Jane Ford, Assistant Professor, and Dr. Virginia Poe, Associate Professor, University of Southwestern Louisiana

The increasing use of microcomputers in the elementary schools requires that elementary children develop keyboarding skills in order to make effective use of these microcomputers. Keyboarding skill requires typing knowledge, which traditionally has not been taught until high school. The implications from research are that children can begin keyboarding skills much earlier (Brady, 1984; Kisner, 1984; Stewart & Jones, 1983).

If keyboarding is to be taught at the lower elementary level, then it seems imperative that the most efficient, most productive, and easiest method should be employed. At present, the keyboard on microcomputers and typewriters is the Qwerty (named for the first six letter in the third row). The arrangement of letters on the Qwerty keyboard was designed to prevent jamming of keys by forcing the typist to slow down (Neill, 1980). With the advent of the "golf-ball" typing element and microcomputers, jamming of keys is no longer a problem (Benford, 1986). A keyboard design based on motion studies was patented by Dvorak in 1936. The arrangement of keys on the Dvorak keyboard reduced movement of the fingers and placed more of the burden of typing on the right hand and the stronger fingers (Neill, 1980). To date, no studies have been reported that compare the use of these two keyboards at the elementary level. Therefore, this study will investigate which keyboard is easier for students in the elementary school to learn and to use.

Analysis of the Dolch Basic Sight Vocabulary List, a popular and frequently used basic sight word list, indicates that on the Qwerty keyboard approximately 34% of the total characters are typed on the home row while 64% of the characters are typed on the home row using the Dvorak keyboard. In the list there are four words that contain hurdles (combinations of letters that require movement of the same finger from top to botton row or vice versa) when typing on the Qwerty keyboard, but no words on the list contain hurdles when typing on the Dvorak. For small, less agile hands, the lack of hurdles and the location of the majority of letters on the home row would indicate greater ease in typing these words when using a Dvorak keyboard. The purpose of this study is to compare the effect of using the Dvorak or Qwerty keyboard on the acquisition of keyboarding skills of elementary children.

ABSTRACT: Keyboarding: Beginning Instruction in a Primary Grade Computer Lab

Ruth Kerr, Hillside School, Allendale, NJ

As the use of personal computers spread, the skill of keyboarding takes on importance that demands attention in the elementary school curriculum. Curricular decisions need to be made as to when to start, and how to implement keyboarding instruction. Hillsdale School in Allendale, New Jersey, decided to include keyboarding in its primary computer curriculum. The presentation will deal with "Keyboarding: the Hillside Way" within the framework of a primary school computer lab.

Background

Hillside School is a public K-3 school of 300 students located in Allendale, New Jersey, a suburb of New York City. In 1980, Hillsdale began utilizing computers in a lab setting rather than individual classrooms. The lab and its curriculum have been continually revised with technological advances. Today it contains 17 Apple computers and four printers. It is staffed by a full-time teacher and 80 volunteers organized by the Hillside PTO. All students are scheduled weekly for hands-on computer use. Special education students, basic skills students, and the gifted-and-talented (Horizon) program are served by the lab.

Keyboarding: The Hillside Way

Four years ago, keyboarding was formally recognized system wide (K-8) as a necessary skill to include in the curriculum. Beginning keyboarding in the primary grades was investigated by attending conferences and workshops, wide reading of current literature in the field, and lots of trial-and-error experiences in the computer lab. The presentation will focus on the resulting keyboarding program now implemented as part of our computer curriculum with specific goals and objectives for each grade level. Included in the curriculum are the introduction of formal keyboarding lessons, as well as methods utilized to introduce the very youngest kindergarten child to the keyboard. Word processing activities used in conjunction with keyboarding activities will be explained.

Through verbal presentations and in written material, specific descriptions of the program and its curriculum will be demonstrated. Included will be practical techniques for keyboarding instruction, training of parent volunteers, and scheduling of a computer lab.

An evaluation of instruction in measurable terms will be presented. General observed outcomes as well as available statistics will be shared.

ABSTRACT: Education Means Business: Project TEACH

Hilary Cowan, Supervisor, Elementary Computer Curriculum, Kanawha County Schools, Charleston, WV

A four-year plan of computer hardware and software implementation, with concomitant teacher training, is currently in place for the 75 elementary schools in Kanawha County, West Virginia. Because the appropriate funding for the first year did not provide for release-time training for teachers, a search for outside financial support was initiated. Project TEACH was created by the Charleston Regional Chamber of Commerce and Development to meet this need. The Chamber has pledged to support the teachers by contributing a combination of visiting speakers, field trips, and money to provide release time for 350 teachers, each of whom received two 3-hour sessions of computer training. The pledge represents a financial commitment by the community of approximately $40,000.

Local businesses have been asked to pledge time, money, or a combination of both, to support the project. Over 80 businesses have responded, and over 85% of their pledges have been a commitment of man-hours in the classroom. Such diverse companies as Union Carbide, Monsanto, IBM, McDonald's, Prudential Bache, and C & P Telephone Company have sent volunteers into the classroom for a three-hour period so that teachers can receive computer training.

The Chamber has hired a person to coordinate the complex job of assigning volunteers to classrooms, usually at the rate of 52 a week. They have also created a training review team whose responsibility is to review the computer curriculum written for the elementary teachers receiving training.

A volunteer committee of teachers and employees worked this past summer to create a Project TEACH kit, which is composed of teaching ideas and tips, child development information, check lists, and sample lesson plans. Voluntary inservices for the business community also have been conducted. Role modeling and ideas for lesson planning help alleviate some of the fears of facing up to 50 children for three hours.

The benefits of Project TEACH far outweigh the complications inherent in such an ambitious plan. Students have been exposed to a variety of topics taught by experts in the field, including geology, insurance, banking, stock markets, carpentry, and music. Teachers are being trained in computer utilization, and business people are gaining a healthy respect for the work of a teacher. Together, businesses and schools are investing in the future of their children.

VIDEO GRAPHICS:
A DESCRIPTION OF THE WORKSHOP AND SOME CLASSROOM APPLICATIONS

Betsy Janssen and Mary Marsh
Albuquerque Public Schools
Computer Education Resource Team
Albuquerque, NM

ABSTRACT

The workshop will demonstrate capturing video images as binary files, saving the files to disk, using several popular software graphics packages to retrieve and enhance the images, and making computer slide shows, video tape presentations, or hard copy exhibits. Hardware will include an Apple IIe, an Apple IIgs, Apple Imagewriter II, a color video camera (portapak), a video cassette recorder with frame by frame pause advance, a large screen monitor, an audio tape player (small "ghetto blaster"), and Computer Eyes color video digitizer interface for the IIGS.

Portrait images can be captured directly from the video camera or images can be taken from a previously recorded video tape. The workshop will address lighting, background, and adjusting the digitizing software for brightness and contrast. The difference between using a computer with an RGB monitor and composite monitor will be shown. The discussion will include selecting cameras, and choosing between saving files in DOS and ProDOS format and deciding what resolution is most appropriate.

After the files are captured, attention shifts to graphics software. Print Shop, Stickybear Draw, Logo, BASIC, and Paint Works Plus are used to retrieve and to decorate the images, which will be saved again with the enhancements, and to print in different sizes. A public domain program for converting files from 33 to 34 format or vice versa will be made available to participants.

Next the files will be presented as a computer slide show. Inexpensive commercial and simple BASIC software will be demonstrated as well as the more sophisticated public domain Cinema slide show program. Cinema uses Terrapin Logo.

A technique for putting the slide shows of images of faces or graphic art on video tape with music accompaniment will pull all the elements of the demonstration together. The resulting tape can be shown anywhere there is a VCR, which may be a more convenient format than setting up a computer to show off student work.

The graphics can be used in a variety of settings. The tapes can be played for Back to School night or any other setting where student work is being shown. Hard copy of the images can decorate notebooks, can be put on filing cards so that substitutes can put names with a faces, can be made into banners, tee shirt transfers, and posters.

Application ideas and management strategies will be included where appropriate throughout the workshop.

This workshop will use only Apple software, hardware, and peripherals, but information about graphics for MS DOS systems and transferring files between the two systems will be available. If time permits, converting graphics files to text files (Run Length Encoded - RLE) will be included.

Although the preceding hardware portion may sound formidable, in fact ComputerEyes is easy to set up and far less frustrating to use than the old 35mm film projectors. Teachers may not be in the habit of using this equipment with their classes, but it is well worth the initial plunge: It is straightforward, and will give students another way to use computers that may become as basic as word processing and data base management. We are assured that in the Information Age, communicating ideas will be an ever more crucial skill.

Opening up computers and inserting interface cards, adjusting cameras, taking pictures, capturing files, making tapes are tasks that can give students a greater sense of control of the technical world that they may not get in other settings. Girls, in particular, often do not have opportunities for these kinds of experiences, and this may increase their self confidence and suggest options they had not previously considered. The material tailored for very young children as well as adults.

Finally, although viewing one's slide show is a rewarding culminating activity, this is *process* not *product* oriented work. It is a project that may involve students in many content areas. The graphics may turn out to be a report on dinosaurs or an editorial about school policy. The presentation may include a talk. It allows teachers to involve students in satisfying in-depth activities.

[Our APS Computer Education Resource Team video graphics workshops have been evolving for several years. Helen Joseph, EQUALS Lawrence Hall of Science, gave us our first ideas. Since then, we (14 of us) have added our own programs, borrowing ideas from discussions with other teachers and from our reading, and enjoyed making presentations in a variety of formats.]

ADULT EDUCATION/VOCATIONAL EDUCATION

Mildred D. Lintner
David Makings
Virginia Owens
Jerry Ice, William J. Seaton

ABSTRACT: A Computer Curriculum for Adult Education

Mildred D. Lintner, Department of Computer Science, Eastern Michigan University, Ypsilanti, MI 48197

The Microcomputer Applications Competency Program, a unique non-credit computer curriculum, is planned specifically to meet the needs of an Adult Education program. The curriculum consists of a structured sequence of nine courses, designed to overcome adult computer fears and to develop today's most needed computer skills in a hands-on environment. Each course helps the student gain both general computer knowledge and high level skills in manipulating industry standard commercially available software, such as *Lotus 1-2-3*, *WordPerfect*, and *dBase III plus*. Written and practical examinations provide progress evaluation. Successful completion of the curriculum leads to a Certificate of Computer Competence.

The presentation includes the following topics: a) goals, objectives and rationale for the MAC Program; b) brief description of the courses, with sample overheads, exercises and training materials; c) description of floor plan, hardware and software for the Microcomputer Teaching Laboratory; and d) evaluation of acceptance and success of the program.

ABSTRACT: Computers in a Work Place Literacy Program

David Makings, Coordinator, Computer Related Technologies, College of Southern Idaho, Twin Falls, ID

Recently, the College of Southern Idaho (CSI) initiated a program integrating the Related Technologies program for on-campus, full-time, vocational students with similar activities offered off campus to part-time, adult students. Related technologies, as used here, refers to the areas of computation, communication, independent learning, critical thinking, and interpersonal relationships - skills so essential in the rapidly changing work place of today.

Computers are used in this program both as a management tool for teachers and as a teaching tool. Word processing systems are used to teach technical as well as personal communications. Spreadsheet packages serve as the basis for numerous types of computational skills from simple algebra to statistics. Students learn to use databases and telecommunications for information searches and as a component of the Life-Long Learning series.

What may be even more unique and exciting about this project is the integration of both on- and off-campus activities under the direction of one program coordinator. Through various activities (including needs assessment surveys of local industries) several sets of related skills competencies will be established. This information will then be integrated into courses offered off campus (either in plant, at a local outreach center, or in the 38 ft. trailer used as our mobile education center) as well as into the more traditional courses offered on campus.

ABSTRACT: Vocational Technical Information System

Virginia L. Owens, PhD, Educational Management and Development, New Mexico State University, Las Cruces, NM 88003

The project is established for the purpose of managing and maintaining a managment information system for vocational education at both the secondary and postsecondary levels in New Mexico. It occurs to me that our integrated approach to fulfilling the goals and objectives of providing a viable database to vocational teachers, administrators, and business people throughout the state would be of interest to conference participants.

The presentation will highlight the following areas:

1. Microcomputer uses
   a. Form design;
   b. Instruction booklets;
   c. Desktop publishing, i.e. brochures, newsletters, flyers;
   d. Correspondence;
   e. Data entry; and
   f. Presentation graphics.

2. Mainframe computer uses
    a. Database creation, maintenance, and storage;
    b. Data analysis;
    c. Report generation;
    d. Data transfer to colleagues through BITNET; and
    e. Electronic communication between our staff and Vocational/Technical Division staff in Santa Fe.

Examples of each of the applications will be included in the presentation.

This project utilizes various software packages available for the Macintosh computer and for an AMDAHL V/8 running the MVS/JES3 operating system. However, all applications, templates, and integrated uses have been developed by project staff.

ABSTRACT: Silicone and Ivy - New Pathways to Higher Education

Jerry Ice, Vice President for Academic Affairs
William J. Seaton, Director, Center for Learning Through Telecommunication

Lecture halls, blackboards, and libraries are having to make room for modems, conferencing systems and on-line databases created by an emerging phenomena - the "electronic college". With growth of moderate cost personal computers, telecommunication between faculty and distant learners, and among distant learners, is changing the very nature of distance education.

During the past year, Thomas A. Edison State College began a pilot program using personal computers to provide distant education to students. Using a limited number of sites, distant learning stations were created which allowed students the opportunity to telecommunicate with faculty and other students. The system, known as CALL - Computer Assisted Lifelong Learning - will be demonstrated with the opportunity for some hands-on experience.

The CALL Network allows faculty to send and receive assignments from students, download the assignments for on-screen grading, and then resend the graded assignments to the student. Students compose assignments and questions on their personal computers, uploading the correspondence for transmission to the faculty member's mail-box when ready. Further, students and faculty can interact on a bulletin board where questions for "discussion" are posted, and students can use the Network's conference component for study groups.

A review of implementation strategies, student and faculty evaluations of the project, and cost data will be presented. Future directions envisioned for the program and a discussion of the issues of importance in telecommunicating will be undertaken.

ARTIFICIAL INTELLIGENCE IN INDUSTRIAL SETTINGS

Gayle J. Yaverbaum
Temple University

ABSTRACT

Professionals within industry and academia are charged with research and development to support the work environment. New processes are being applied that incorporate intelligence. This session addresses the industrial world and describes some current efforts within the field of artificial intelligence applying intelligent interface design techniques and expert systems.

Shelly Dews, NYNEX Corporation

An intelligent interface aids users in their interaction with a computer. For most computer users, the primary goal is not to use a computer but rather to do a job using a computer as a tool to allow the job to be done more efficiently. An intelligent interface is designed with that goal in mind. It provides a way for users to do work in a manner consistent with their current work habits.

At NYNEX, an interative design process is being used to design and develop intelligent interfaces. Shelly will discuss this process which includes analyzing the system tasks and users, prototyping an interface, evaluating it, and redesigning it to incorporate the results of the evaluation. Currently intelligent interfaces for a network management work station and an object-oriented network behavior simulation are being developed.

Stephen Breslawski, Gayle J. Yaverbaum, Temple University

This presentation will describe the architecture for a Technical Intermediary Expert System (TIES) model that will help a manager (a) identify a problem, (b) formulate the problem so that a system can correctly interpret the problem, and (c) creatively combine tools for problem analysis.

TIES serves as a consultant for managers in their quest for information and further provides the necessary education to accurately interpret appropriate tools. The system will theoretically save time and lower the cost of expertise needed for problem solving activities.

Dale Hankins, Reinhard Ziegler, Arthur Andersen & Company

Training applications that take advantage of artificial intelligence technology are still relatively scarce. Arthur Andersen & Company believes that AI has extensive possibilities in a host of training/education environments. As leaders in Systems Integration, particularly in the manufacturing industry, they are seeking to apply Just-In-Time (JIT) business concepts to manufacturing training. AI is an outstanding tool for making an "expert's" decision process available to the less experienced worker. When AI is configured with interactive training tools, the inexperienced worker can not only diagnose a problem, but can also have instant access to the right training content, presented at the right level of detail via the right textual, visual and/or graphic media at the right time.

The panelists will discuss the instructional design issues that surround business and industrial applications. Two specific projects that are currently being designed and developed will be presented.

PANELISTS

Shelly Dews
NYNEX Corporation

Stephen Breslawski
Temple University

Dale Hankins & Reinhard Ziegler
Arthur Andersen & Company

POTPOURRI

Katherine Pearson-Jagoe
Robert Price
Susan N. Kirkpatrick, Barbara Biglan

ABSTRACT: Association for Higher Education of North Texas, Pilot Project for a Union Catalog on CD-ROM

Katherine Pearson-Jagoe, Director of Library Programs and Services, Association for Higher Education of North Texas

The Association for Higher Education of North Texas (AHE) has received an LSCA Title III Grant for fiscal year 1988 from the Texas State Library to develop a union catalog of the holdings of five major Texas libraries: Baylor University, Dallas County Community College District, Dallas Public Library, The University of North Texas, and The University of Texas at Arlington. Optical disc technology is being tested to determine if it is a feasible medium to use for a union catalog of this magnitude. The database consists of approximately 1,000,000 unique records for materials owned by these institutions. The type of optical disc being tested for this project is Compact Disc-Read Only Memory.

AHE is working in cooperation with the AMIGOS Bibliographic Council to obtain the completed CD-ROM Catalog. AMIGOS has prepared the database by using the existing machine-readable records of the five institutions and creating a unique list of all titles owned and the name of the libraries owning them. AMIGOS has also provided the CD-ROM product and software called the *Intelligent Catalog*, developed by the Library Corporation. This software combines hypertext capabilities and expert systems technology to optimize the ability of the library user to find materials in a variety of ways.

Special features of the software include natural language searching; Boolean searching; maps of the campus and library floor plans; ability to save searches and print bibliographies by author, title, or call number; spoken tutorials; record of search paths used; and suggestions provided for other items in the database that are related to saved searches that may be of interest. Local information about events can also be accessed on screen by the user. The Union Catalog is updated on a monthly basis using the hard disk that is included in the workstation.

ABSTRACT: Producing Interactive Video Programs on a Limited Budget

Robert V. Price, Assistant Professor, Program Director for Instructional Technology, Texas Tech University, Lubbock, TX

The interactive videodisc offers the potential to combine high quality motion, still frame, and text images on an optical disc, which can store up to 54,000 individual frames of information that can be controlled by a computer program for individualized instruction or can be used with a videodisc player alone for group instruction. The potential of the videodisc medium to improve instructional practices by providing teachers and students with easily accessible visual databases is enormous.

While production of videodisc-based instructional programs is usually very expensive, the use of new and improved nonbroadcast quality video equipment, standard 35mm cameras, and DRAW technology videodiscs allow interactive video programs to be produced on a small budget.

Several original videodisc-based instructional programs have been produced by the College of Education at Texas Tech University on very small budgets. The production team, consisting of graduate students working under the direction of a faculty member, used video and still camera equipment previously owned by the college. Planning tools included pert charts, instructional design flowcharts, and scripts. Media production was done on campus and on location using portable equipment. Production of the final videotape masters for the programs has been done using both standard 3/4" video equipment and 1" equipment available on campus. The production of a "check" videodisc from the master tape was contracted to a commercial videodisc production house. Several computer-assisted instruction programs using both locally made videodiscs and pre-existing generic videodiscs have been produced. The programs have been written for the Apple II computer using BASIC, Pascal, and the Laserwrite authoring system.

One pilot videodisc program concerned teaching the language experience approach to reading. The program was pilot tested in the summer of 1987 with a control and experimental group of students and was found to be equally as effective as the conventional instruction while resulting in both instructor and student time savings and positive student attitudes. Informational programs on the college have also been produced as well as several programs which use pre-existing generic videodiscs.

A Texas Tech-produced generic videodisc on computing technology is scheduled for release in June 1988. This disc will be marketed by an independent company as a teaching resource for courses in data processing, computer science, and educational computing. CAI programs for Apple II, IBM PC, and Macintosh computers will also be produced for the computing technology videodisc.

The Texas Tech experience has proven that videodisc technology is now within the reach of instructional programs with limited budgets if basic media and computing resources and talent are available.

ABSTRACT: Artificial Intelligence in the Classroom

Susan N. Kirkpatrick, Mathematics Department Chairman and Director of Computer Program, Shady Side Academy Middle School, Pittsburgh, PA; and Barbara Biglan, Director of Computer Program, Shady Side Academy Junior School, Pittsburgh, PA

Until recently, Artificial Intelligence has been reserved primarily for use in medicine, business, law, and industry. Educational research at the university level has produced functional intelligent tutoring systems, but AI has seen limited application in precollege education. This presentation will relate an AI project which is in its third year at Shady Side Academy. It will offer ways in which this highly complex topic can be "watered down" for secondary and elementary classrooms.

In conducting this project, involving the use of expert systems and robotics, the goal is not to create the ultimate, "intelligent" machine, but to (a) explore problem solving processes both algorithmically and heuristically, (b) study simple programming structures of expert systems, (c) discuss the use of expert systems in industry, business, law, and medicine, and the application of the expert systems to enhance education, and (d) build and program a functional educational robot.

There are three parts to this project. Part 1 deals with the programming aspects of expert systems. All students must have knowledge of programming in BASIC and Logo, and advanced students will be introduced to the traditional AI languages. Since the best designed expert systems are written in PROLOG or LISP, the goal is not to create a truly efficient expert system, but to explore its structures. The differences between algorithmic and heuristic programming will be explored here. Part 2 deals with the applications of expert systems. Students research and discuss the use of expert systems in business, industry, medicine, and law. At local universities students actually use expert systems, in essence, intelligent tutoring systems, for education. In Part 3, students build a robot which can be interfaced with a computer and programmed to do simple tasks. The relationship between robots and expert systems is discussed, as well as language processing, visual processing, and robot locomotion.

EDUCATIONAL COMPUTING IN MINORITY INSTITUTIONS

Jesse C. Lewis, *Chair*
Vice President for Academic Affairs
Norfolk State University

ABSTRACT

This session will examine the current status and projected directions of some aspects of educational computing at minority institutions. Issues addressed will focus on curricula, computing resources, professional development, and accreditation. Concerns of persons in the audience regarding computing in their respective environments will be addressed during a question and answer period.

Where Do We Go From Here?

Forty years ago, some of the greatest minds of this nation were asked to predict the most important inventions for th upcoming ten years. They missed both the atomic bomb and electronic computers. I dare in this presentation to discuss future directions in educational computing with the same probability of gross error. There are two primary reasons I can afford to be foolhardy, however: (1) I do not have to take the responsibility since I am giving a composite summary of the opinions of more than three hundred college educators; and (2) as is the case with all predictions, by the time your theses are proven false it is too late to matter. As we begin there is a need to document some of the hypotheses on which the theses are based.

Academic and Administrative Computing: Where Should the Twain Meet?

The computer revolution has had a major impact on all areas of society. In particular, it has provided new challenges and exciting opportunities for educational institutions. It has led to improvements in administrative operations and enhancement of the institutional process. Across the country, colleges and universities have mobilized resources in an effort to exploit the benefits of computer technology. Many institutions, however, fore a variety of reasons, have not been able to adequately reap the benefits of effective campus computing. This presentation will provide some suggestions and observations that may aid institutions in planning and implementing an effective program of campus computing.

Computer Research in Minority Institutions: It's Alive, Well, and Profitable for the University

There are organizations that are willing to encourage research at historically black colleges and universities by funding such activities; experience in research is very important to the future welfare of minority students; minority institutions must compete with highly funded majority institutions and yet keep their identity and mission. All of these ideas are presented as reasons why in many cases historically black colleges and universities are doing very well in computer research. Not only do research funds enhance the ability of universities to keep up with technological trends, they also provide unique opportunities for students and enhance the national reputation of the faculties.

Preparing for Accreditation

National efforts to accredit baccalaureate degree programs in computer science are underway. The accreditation criteria used for program evaluation establish minimum standards that all graduates form a program must satisfy. While the decision to seek accreditation is presently a voluntary option, the widespread acceptance of the need for accreditation indicates that this will not remain the case. Thus, in order that minority institutions remain competitive in the computer science discipline, accreditation is an issue that must be addressed. This presentation will focus on the program requirements—curricula, laboratory resources, faculty, and students—necessary for accreditation as well as some strategies for preparing for a successful accreditation visit.

PANELISTS

Nellouise D. Watkins
Bennett College
Greensboro, North Carolina

William B. DeLauder
Delaware State College
Dover, Delaware

George C. Harrison
Norfolk State University
Norfolk, Virginia

Sandra J. DeLoatch
Norfolk State University
Norfolk, Virginia

SPONSOR: ECMI

CAI AND EQUITY: K-12

James Burney, Michele Walker
Alison Elliott
Brooke Woods
Anne Pitts, Judine Johnson, Barbara Simpkins

ABSTRACT: Development of CAI Tutorials to Accompany the Alabama Basic Competency Testing Program

James D. Burney, Codirector, and Michele Walker, Programming Coordinator, Alabama Educational Computing Research and Development Network, University of North Alabama, Florence, AL 35632

Since 1984 educators from the School of Education at the University of North Alabama and engineer/programmers from the Johnson Research Center at the University of Alabama in Huntsville have worked together to develop tutorials keyed to the 199 competency statements included in the Alabama Basic Competency Testing Program. Additional participation by the Alabama State Department of Education and 14 local school systems as consultants and field test sites has made the project a statewide effort. Competencies are stated for third, sixth, and ninth grades in mathematics, reading, and language arts. Each competency has one or more programs designed to allow independent study by students who fail to meet minimum standards on a statewide standardized test. Programs are mastery-based, with pretest, tutorial, practice, and posttest sections. All materials are for the Apple II series of computers, and are written in high-resolution graphics using custom fonts and a proprietary authoring system. Extensive branching, looping, and animation are included. In selected reading and language arts programs where auditory stimulus is mandatory, voice synthesis is incorporated using the Echo voice synthesis card. An example of this is the first reading competency where students must recognize 200 basic Dolch sight words.

The Network has currently distributed over 120 programs to schools in Alabama. All mathematics competencies are complete, and about 20 reading and language arts competencies are either distributed or are ready for distribution. All school systems in Alabama now use the materials, and feedback has been extremely positive. Five distribution centers have been organized throughout the state to insure that current editions of all programs are locally available to teachers. Each of these centers maintains the current software, copying facilities, and all documentation. The software is free to any educator in the state. Teachers are encouraged to share their copies with others. This has resulted in widespread dissemination throughout Alabama.

ABSTRACT: Sex Typing of Young Children's Behaviours in a Computer-Active Preschool Classroom

Alison Elliott, School of Policy and Technology Studies in Education, Faculty of Education, The University of Wollongong, New South Wales, Australia

Early childhood teachers in Australian preschools are becoming interested in the possibility of adding computer based learning experiences to the range of more traditional activities already offered in their centres. The study outlined in this project presentation attempted to address some of the curriculum issues relating to the use of computers in Australian preschool classrooms. The major aim of the study was to explore young children's behaviours while participating in developmentally appropriate computer based activities in preschool classrooms and to make recommendations to early childhood educators on matters of policy and curriculum. Because issues relating to gender and equity are of concern to early childhood educators, an important part of the study was to compare aspects of the behaviours of girls and boys in the computer area. The findings reported here represent only one dimension of the study.

The subjects of the study were four-year-old children enrolled in four Sydney preschool classes who attended preschool for two or three sessions each week. Before the computer was introduced to the children, information was gained about their attitudes toward, knowledge of, and previous experience with computers. Children were also asked to rate their interest in a number of play activities, including playing with the computer.

In this study the computer was set up in a learning centre and available to the children during the "free play session." Children could choose to work in the computer area just as they would choose to work in any other learning centre. After a period of familiarity with the computer, their behaviours were observed and recorded as they used two developmentally appropriate activities: a drawing program and a single-key stroke version of Logo, with a floor turtle. Information was collected on the frequency and duration of their computer related play, and on their playstyle and interactive behaviours in the computer area. Observations in each classroom took place over a six-week period.

The purpose of this presentation is to highlight some of the preliminary findings of the study that indicate gender-related differences in children's participation in computer activities and in their behaviours in the computer area. Measures of children's computing knowledge and attitudes, and their expressed interest in playing with computers also revealed significant gender-related differences. The findings are of interest and concern as they suggest that girls and boys are already seeing the use of computers in society and the classroom as a male-oriented activity. The possible sex typing of computing at such an early age has several important implications that relate to the development of differential skills, attitudes, outlooks, and future goals. For early childhood educators the need to take action to encourage more gender-equitable participation and experience in computer play would seem to be an essential requirement of curriculum planning and practice.

ABSTRACT: Computer Technology in an Extended Time Summer School Program

Brooke P. Woods, Director of Special Projects, Jackson Public School District, P.O. Box 2338, Jackson, MS 39225

Believing that there is a core of basic knowledge that all children are capable of learning, and yet that a variable factor does exist in the children's rate of learning, the educators in the Jackson Public School District provide for children needing additional time and instruction in order to master objectives essential for promotion to the next grade level in the means of an Extended Time Summer School Program.

Traditionally the Extended Time Summer School Program has consisted of instructional delivery by a teacher and a teaching assistant with no provision for the use of computer technology in the summer school curriculum. The introduction to the computer for students in the district's thirty-seven elementary schools (grades 1-6) has occurred during the school year via a small lab of six microcomputers in each school.

The summer of 1987 witnessed the blending of the two programs; for the first time computers were used as vehicles with which to assist in accomplishing the objectives of the Extended Time Summer School Program. By moving the computer labs from the sites where summer school was not held into the eight sites where summer school was held, it was possible to install a Computer Learning Laboratory with a total of eighteen computers in each of the designated summer school sites.

The purpose of this Learning Lab was to serve as a reinforcement for the instruction delivered by the classroom teacher. A resource teacher was hired to manage each Learning Lab. The role of this teacher was to guide the students as they used computer software to work through lessons in their needed areas of improvement. The identified software, which showed a high correlation with district learning objectives, was such that a student could begin at his/her own particular level and progress at his/her own rate.

Approximately 225 students in each of the eight elementary school sites spent one and one-half to two hours per week in the Computer Learning Lab. Students who had been labeled as low-functioning anxiously awaited "computer day" as they began to comprehend concepts that they previously had not grasped. As a result of new accomplishments, feelings of triumph and self-worth began to surface.

The project received high marks not only from the students, but from parents, teachers, principals, and district administrators alike. The use of the computer as a learning resource in the summer instructional program brought about higher attendance, improved behavior, and an increased motivation toward learning. Not only was there a verbal message of success from all involved, but written evaluations from teachers and administrators were positive, and the pretest and posttest data gathered from the project evidenced an average gain per student of more than five percentage points.

ABSTRACT: Integrating the ECIA Chapter 1 Program at the District of Columbia Public Schools

Anne W. Pitts, Executive Director; Judine B. Johnson, Deputy Director; and Barbara D. Simpkins, CAI Coordinator; ECIA Chapter 1 Program

The Chapter 1 Program of the District of Columbia Public School serves approximately 16,000 students. The goal of the program is to improve the performance of the eligible population in reading, writing, and mathematics. The Computer-Assisted Instruction Program (CAI) both at the elementary and junior high school levels, comprises the major instructional intervention for Chapter 1 identified students. Grades 4-6 receive CAI in the elementary school, while Grades 7-9 are provided CAI in the junior high schools. Chapter 1 services are offered as a supplement to the services given by the regular budget program of the school district. Therefore, Chapter 1 instruction is integrated with the school system's Competency-Based Curriculum.

"The Collaborative Process" is the key strategy for integrating Chapter 1 CAI with the regular budget instructional services. This model is implemented at the local school level to ensure ongoing conferencing between the Chapter 1 and the classroom teacher on the individual educational plan for each student whom they both teach. The model also utilizes a diagnostic-prescriptive teaching approach. The three-step collaboration process allows the two teachers to jointly (a) assess the needs of the student through analyzing the student's standardized test results, computer/classroom progress records, cumulative records, and the student progress plan; (b) planning the instructional format, including the objectives, teaching strategies, and learning activities, and (c) implementing a well-coordinated instructional program in both the Chapter 1 laboratory and regular classroom with ongoing conferencing on student progress.

The Chapter 1 computer-assisted instruction courseware has been carefully correlated with the curriculum of the school district. Courseware selection has been made on the basis of its capability to match the objectives and skills that are a part of the "Student Progress Plan." Additionally, in support of the school district's writing initiative, a CAI writing program has been incorporated in the Chapter 1 program to complement the district's curriculum. At the junior high level, in particular, teachers, administrators, and students have provided input into the courseware development.

The ECIA Chapter 1 Elementary CAI Program was cited by the United States Office of Education as an exemplary project primarily because of the effective integration of CAI with the school district's curriculum.

ONE SCHOOL SYSTEM'S STRUGGLE TO ACHIEVE THE HIGHEST LEVEL OF COMPUTER INTEGRATION

Jenelle V. Leonard
Joseph R. Harris
Shirley H. Bellamy
et al.
District of Columbia Public Schools

ABSTRACT

The District of Columbia Public Schools is a large urban school district with approximately 90,000 students, 200 schools, and over 6,000 instructional employees. As part of its instructional computing effort, a Computer Literacy Five Year Plan was implemented beginning in 1983. The first major task of this plan was the establishment of a student computer laboratory in every school. After having accomplished this, the next major goal was facilitating the utilization of computers by teachers and students.

Initially, the Computer Literacy Training Laboratory, a centralized staff training facility, provided training for the computer-using teachers on basic computer literacy essentials such as hardware operations and educational applications. All instructional personnel were required to take an introductory computer literacy course as part of their license renewal requirements. Training on software review evaluation procedures was also provided to assist schools with the selection of software to meet their individual instructional needs. Once software had been selected and acquired, additional training was also provided on its use in the classroom.

In spite of all training provided to the teachers, it became evident to those individuals responsible for implementing the instructional computing program that the use of computers in the local schools was not a natural evolutionary process. Numerous meetings, debriefing sessions, and discussions occurred to ascertain and identify why the computer integration process for teachers was not as natural as one would expect. Implementation and integration models for other innovations in the school district were even examined in an attempt to resolve the problem.

The missing ingredient/component that was finally identified was the lack of integration between classroom subject-area curriculum and computer-based instructional software. Teachers, already intimidated by using computers, were frustrated by having to learn how to operate different types of software programs and to relate the software to what they were doing in their classrooms. The solution was to provide learning activities and lesson plans that were correlated to the various content areas and that utilized the technology.

Once the solution was determined, DCPS assembled subject area curriculum writers and computer specialists to begin the development of instructional materials and computer-based activities which would facilitate the integration process. Since many of these persons were full-time employees assigned to local schools during the regular school year, most of the effort took place during the summer months. Initial efforts progressed slowly due to the massive nature of the task. In order to resolve this, DCPS entered into a unique collaborative effort with a publishing company to develop a truly integrated instructional program.

As the collaborative development process progressed, a software decision was made which has turned out to be a major benefit of using the integrated curriculum materials. At that stage of the process, office productivity software (i.e., tool application software such as word processors, spreadsheets, and databases) was emerging as the primary use of microcomputers in the business environment. With the desire to give all students experience with these tool applications, it was decided to use these tool applications as vehicles for the computer-based subject-area instructional activities under development. For example, mathematics activities could involve spreadsheets, social studies could take advantage of databasing, and language arts could promote writing skills through word processing. An added benefit of this approach was that it would also minimize the types of software that the classroom teachers would be required to operate and would greatly relieve an already inadequate software budget.

The collaborative effort between DCPS and the commercial publisher turned out to be the key to the success of the entire computer and curriculum integration project. DCPS provided the curriculum writers and computer specialists to develop the instructional scope and sequences and instructional kernels for further development. The commercial publisher provided the necessary staff and development resources to finalize the product. Twenty-two additional school districts also participated in the collaborative effort by contributing computer-based instructional activiites. The final product is called MicroWorks and materials have already been completed for grades seven through nine in mathematics, science, social studies, and language arts with plans underway to develop materials for other grade levels and subject areas. DCPS is proud of its contribution to the true integration of computers into the teaching and learning environment.

ARTIFICIAL INTELLIGENCE PROGRAMMING: A STYLE FOR MEANINGFUL STUDENT ASSIGNMENTS
A VERSION OF BLOCKSWORLD AS AN AEXAMPLE

William R. Adams
Assistant Professor
Department of Computer Science
University of New Haven
West Haven, Connecticut 06525
(203) 932-7064

ABSTRACT

Programming assignments for any computer science course pose the problem of being either trivial, but realistic assignments, or meaningful learning experiences which are not realizable within the constraints of time or the ability of students in a one semester course. LISP, however, if used as the language of choice in an Artificial Intelligence (AI) course, offers a workable escape from this dilemma.

A "LADDER AND RUNG" method of LISP program design is presented as a successful solution to this problem. This method allows a major AI LISP program to be designed with one, or even a few, RUNGS missing from the LADDER of the program. These RUNGS correspond to code modules which are assignable to individual students or student groups.

This method of conceptualizing program design, permits assignment emphasis where it belongs, that is, on the topic of concern and not the bookkeeping code that can often clutter an otherwise clean-bit-of-code and thus cloud the image of understanding that students must build for real learning.

As a means of demonstrating this method, a classic AI programming assignment will be explained and defined as a ladder and rung structure. The program defined will be a variation of a BLOCKSWORLD program [1].

INTRODUCTION

The rapid expansion of computer science programs at the university level has pressed faculty into developing new courses rapidly in order to keep up with state-of-the-art technology. Often the course material treats rather specialized topics in the computer science domain. Faculty developing syllabi, very often resort to designing courses around a theoretical frame and removing students from practical, real programming experiences. This tendency is understandable in light of the fact that the topics being studied are often a small (code-wise) component of any possible application. Assigning a complete application program in order to give students a practiced familiarity with some small, but important, code structure has been unrealistic and in most cases counter productive.

Artificial intelligence (AI) courses are particularly prone to being taught at the purely theoretical level because of the many and diverse nature of the topics. Further consideration arise after feedback from industry has indicated that practical programming experience is an important component of what they expect from CS/AI graduates. Realizing the problem of balancing real world expectations with academic constraints, the following ladder and rung method of program conceptualization is presented. The various methods are not new and the techniques have been available to experienced LISP programmers as long as LISP has been around. However, by coalescing several techniques and ideas into an integrated whole and giving this composite the distinction of a name, we will hopefully stimulate its use as a powerful solution to programming assignment problems.

BASICS OF THE LADDER AND RUNG STYLE

LISP programming consists of defining functions. Functions are typically, small general purpose modules of code with a well defined parameter passing structure, a conditionally constrained functionality and, in LISP, a definite required return value.

Programs, as such, in LISP, are functions calling other functions (or themselves recursively) passing the expected parameters or arguments for the code to operate upon in order to return (in its calling position) some functionally manipulated value. This consistent structure of:

1 - a requisite input structure,
2 - an internal functionality, and
3 - a well-defined output structure,

allows major LISP programs to be thought of as a ladder structure where the rungs of the ladder are the several functions which, as a whole, constitute the program.

This style of programming is better known as modular programming and is often used in well designed programming projects. In modular programming, different groups work on different parts of the larger problem and bring their individual efforts together at some integration phase of project development. This style of program development works well, albeit slowly, in real life settings. In the academic setting the student, as the module programmer, does not have the time, or we as the instructors, do not wish him to expend his time in designing front-end and response-end modules in order to test the assigned module for functional accuracy.

We, as instructors, would better serve our purpose by providing the ladder framework with only the functionally critical rungs missing so as to direct the student into focusing his attention on the "theoretical topic" of concern. The ladder and rung method, also, encourages the student to experiment beyond the requirements of the assignment in that he may redesign other rungs of the ladder in response to his own dictates and not concern himself with every aspect of the complete program.

AN EXAMPLE USING A VERSION OF BLOCKSWORLD

Early in the teaching of artificial intelligence procedures, one introduces the concept of searching, that is, the process of testing possible configurations of a problem (permissible states) until some previously defined solution state is found. The uninitiated are referred to [2] for a complete explanation of search methods.

The BLOCKSWORLD program configured for my students is defined in the following manner.

A robot (fictitious or real) is assumed to understand (is able to perform) four basic block moving commands:

PICKUP(X) - Block X is lifted from its table height position to some position above the table top.

PUTDOWN(X) - Block X is placed on the table.

UNSTACK(X Y) -Block X is lifted from its position on block Y to some position above the table top.

STACK(X Y) - Block X is placed upon block Y.

(The project of converting these high-level commands into robot arm microcode is another topic of AI programming.)

Initially the program BLOCKSWORLD, is presented with a configuration of blocks such as shown in (fig.1), (actually some symbolic representation of this configuration). This is known as the Initial State. Then the program is presented, in like manner, with another configuration, the Goal State such as (fig.2). The solution of the problem is the finding of the PATH, (the list of moves) which would realize the transformation form initial to goal state.

A pictorial representation of these states is not stored, but some unambiguous symbolic description must be devised for the internal representation, for example:

Initial state definition -
```
  ( (ONTABLE (A))
  (ON (E A))
  (ON (C E))
  (CLEAR (C))
  (ONTABLE (D))
  (ON (B D))
  (CLEAR (B))   )
```

Goal state definition -
```
  ( (ONTABLE (E))
  (ON (A E))
  (CLEAR (A))
  (ONTABLE (B))
  (ON (C B))
  (CLEAR (C))
  (ONTABLE (D))
  (CLEAR (D))   )
```

An important topic in AI is the method by which the search is conducted. This is analogous to asking how we, as humans, search for problem solutions. As Winston [2] points out, searches can be directed or blind. Blind searches are nothing more than random walks through some state space (the collection of possible configurations) of the problem until happening upon the goal state. This search method, obviously, lacks an "intelligent" component. The antithesis of blind searching is directed searching. AI's role in this type of problem solution is to define the method by which the search can be directed (the "intelligent" component). This important, but code-wise minor, component of the BLOCKSWORLD program, is known as a 'state evaluation function'. In AI, the design of this evaluation function is of paramount concern and should be stressed in instruction, practiced in exercises and tested by trial. This is a prime example of where the code for the complete application program far outweighs the code of interest and one is tempted to forego the programming assignment.

Looking at a schematic diagram (fig.3) of the complete BLOCKSWORLD program (the ladder) we see the many separate components (the rungs) which are required.

The importance of the ladder and rung method is not only in its modularity, but in the way the modules are prepared and presented to the student.

As illustrated in the Blocksworld program there are six different rungs or modules which comprise the Blocksworld ladder, and five rungs to the associated DEPTH_FIRST_SEARCH ladder. The blocksworld ladder is described to the students, much as was done here. The students are further given a software copy of the ladder which has certain functional rungs replaced with assignment rungs. These assignment rungs are named as per their functional name and will be called in place of the function call. The student runs the ladder program and interacts as required until the assignment is presented. In Blocksworld the first assignment rung usually encountered is STATE_DEF_IN. Here the student is directed to code a module to conform to the description given under the function name. As the student defines this new function definition in the working environment, the assignment module of the same name is deactivated. As the student completes this module the Blocksworld ladder program continues to the next assignment module and the student either proceeds or awaits further class instruction if the assignment so requires.

In the Blocksworld example, there are also dummy function calls. The DEPTH_FIRST_SEARCH assignment was described as a blind search. Students soon discover the futility of this search method and wish to improve program performance. The EVALUATION function can be included as this type of do-nothing dummy call. The next assignment then becomes the writing of heuristic evaluation functions.

This method of assignment presentation can be compounded to whatever level the topic demands.

OBSERVATIONS

I have presented this method of assignment to numerous Artificial Intelligence classes, both at the undergraduate and graduate level, and I have made the following observations:

1) Students do not seem to experience the "I can't get started!" syndrome, because the program is already

started!

2) Students work on the code assigned in order to see the program progress. Often they code and recode the assigned section several times, making improvements with each go around.

3) Students tend to guard "their" program and there is less sharing of code.

4) Students take more pride in their unique solutions to the assignment.

5) Students often add or modify other rungs of the ladder to reflect their own personalities and thus gain further self-motivated programming practice.

6) Students compete not only with themselves, but try to out-do the ladder program they were first given. They compete with the instructor.

CONCLUSIONS

The ladder and rung method of programming assignment is a very effective modified version of modular programming. The style forces careful planning of programming assignments and requires that complete application programs be designed, coded and then modified to include assignment modules. The effort necessary, before assignments are given, ensures that student make the most effective use of their time, learning first those topics that we designate as most important. The student is not discouraged from attempting major changes and/or improvement of the ladder program because he can stop at any time and revert to the existing structure if time considerations dictate. The ladder structure is also easily modified to stress different points of interest. The additional effort required before instruction guarantees that students learn in a more orderly and effective manner.

BIBLIOGRAPHY

1. Winograd, Terry, *Understanding Natural Language*, Academic Press, New York, NY (1972).

2. Winston, Patrick Henry, *Artificial Intelligence*, Second Edition, Addison-Wesley, Reading, Ma (1984).

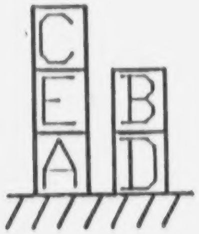


FIG. 1
INITIAL STATE

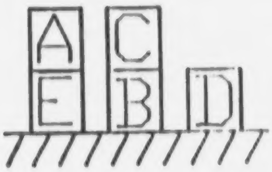


FIG. 2
GOAL STATE

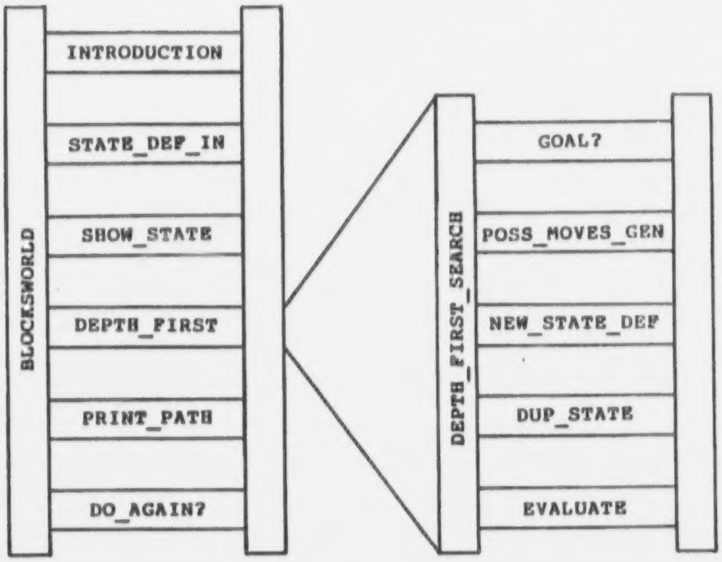


FIG. 3
BLOCKSWORLD FUNCTIONAL DIAGRAM

Blocksworld functional diagram.

```
Ladder - Name...............BLOCKSWORLD
         Rungs on ladder....#1 - INTRODUCTION
                            #2 - STATE_DEF_IN
                            #3 - SHOW_STATE
                            #4 - DEPTH_FIRST
                                 #4A - GOAL?
                                 #4B - POSSIBLE_MOVES_GEN
                                 #4C - NEW_STATE_DEF
                                 #4D - DUP_STATE
                                 #4E - EVALUATE
                            #5 - PRINT_PATH
                            #6 - DO_AGAIN?
```

Rung 1
Function name.....INTRODUCTION
Input parameters..none
Module function...List the purpose of the blocksworld program and display any instructions for the user.
Return value......True.

Rung 2
Function name.....STATE_DEF_IN
Input parameters..literal atom-name of state definition being input.
Module function...User interactive module requesting information needed to build symbolic state definition.
Return value......Symbolic state definition list.
ex: Using fig.1...

```
    INITIAL_STATE_DEF =
                     (  (ONTABLE (A))
                        (ON (E A))
                        (ON (C E))
                        (ONTABLE (D))
                        (ON (B D))
                        (CLEAR (C))
                        (CLEAR (B))       )
```

Rung 3
Function name.....SHOW_STATE
Input parameters..Symbolic state definition.
Module function...A graphics display of the state represented by the state definition passed in.
Return value......True.

Rung 4
Function name.....DEPTH_FIRST
Input parameters..A queue or list structure defined as follows.

```
ex: ( element1  element2 ... elementN)
where: element is...
((current_state_def)
 (possible_moves list)
)
```

Module function...First - check if current state is goal state.
Second -if not goal, generate next state

applying first possible move on list to current state and add to front end of queue with this state's possible moves if it is not a duplicate state. If duplicate state go to second move on list and try again.

Third - Recurse until goal state found or queue empty.

Return value......Queue as described above.

Rung 4A
Function name.....GOAL?
Input parameters..Current state definition.
Module function...This function compares its input parameter against the GSTATE global variable.
Return value......1.) True if the current state matches GSTATE.
2.) NIL otherwise.

Rung 4B
Function name.....POSSIBLE_MOVES_GEN
Input parameters..The current state definition list.
Module function...Generate an unprioritized list of moves which are possible from the current state.
Return value......A list of possible moves. Example...
((PICKUP (C)(PICKUP (B))) see fig.1.

Rung 4C
Function name.....NEW_STATE_DEF
Input parameters..1.) Current state definition.
2.) Move. Ex: (PICKUP (C))
Module function...By means of some production rules modify the current state definition by applying the move and generate the next state.
Return value......A new state definition.

Rung 4D
Function name.....DUP_STATE?
Input parameters..1.) Existing queue.
2.) New state definition.
Module function...This function checks through the queue for a previous occurrence of the new state definition.
Return value......1.) True if this is a duplicate state, ie: one already found on the queue.
2.) Nil (false) if not duplicate state.

Rung 4E
Function name.....EVALUATE
Input parameters..1.) Goal state definition.
2.) Possible moves list.
3.) Current state definition.
Module function...This function is to prioritize the list of possible moves putting the most likely move on the front of the list.
Return value......The input list of moves, but reordered in some priority order.

Rung 5
Function name.....PRINT_PATH
Input parameters..The queue.
Module function...Process the queue in reverse order collecting the moves which affected the reconfiguration of the blocks. Print out the list of moves.
Return value......True.

Rung 6
Function name.....DO_AGAIN?
Input parameters..None.
Module function...This function determines, by prompting the user, if the BLOCKSWORLD program is to be run again or if it is to be exited.
Return value......NIL.

```
              ~Appendix 1  -  BLOCKSWORLD ASSIGNMENT CODE
              ~-----------------------------------------
~*************************************************************************
~CODE WRITTEN IN IQ LISP
~*************************************************************************
  ~main function of BLOCKSWORLD  program

(DEFUN BLOCKSWORLD (LAMBDA ()
  ~select 25 X 80 text mode
(DISPLAYMODE 80)
  ~center string on row 1 on a 80 column display
(PCENTER 1 80 "THE BLOCKSWORLD PROGRAM")
  ~print text. Begin on row 3 and format for 80 columns
(PTEXT 3 80 ASMT_BW_TEXT)
(PCENTER 24 80 "PRESS ANY KEY TO CONTINUE WITH INITIAL STATE DEFINITION.")
  ~clear the keyboard buffer
(CLRBFI)
  ~wait for keystroke
(TYI)
  ~begin an iterative loop for data input
(PROG ()
  ~define label for goto
REDO_INITIAL_STATE
  ~assign initial state definition to variable name ISTATE
  ~*STATE_DEF_IN is first student function
(SETQ ISTATE (*STATE_DEF_IN "INITIAL STATE"))
  ~graphic display of initial state as defined by user
(SHOW_STATE ISTATE)
(PCENTER 24 80 "ACCEPT STATE PRESS A, REDEFINE STATE PRESS R")
  ~clear keyboard buffer
(CLRBFI)
  ~read response and continue or loop to REDO_INITIAL_STATE label
(COND
  ((EQ (TYI) "R")(GO REDO_INITIAL_STATE)) )
  ~define label for goto
REDO_GOAL_STATE
  ~assign goal state definition to variable name GSTATE
  ~*STATE_DEF_IN function is reused
(SETQ GSTATE (*STATE_DEF_IN "GOAL STATE"))
  ~graphic display of goal state as defined by user
```

```
(SHOW_STATE GSTATE)
(PCENTER 24 80 "ACCEPT STATE PRESS A, REDEFINE STATE PRESS R")
  ~clear keyboard buffer
(CLRBFI)
  ~read response and continue or loop to REDO_GOAL_STATE label
(COND
  ((EQ (TYI) "R")(GO REDO_GOAL_STATE)) )
  ~assign search path list (queue) to variable QUEUE
  ~call *DEPTH_FIRST, the second student assigned function
  ~initialize QUEUE with ISTATE
(SETQ QUEUE (*DEPTH_FIRST (LIST ISTATE)))
  ~print path from QUEUE
  ~call *PRINT_PATH,  the third student assigned function
(*PRINT_PATH QUEUE)
  ~allow for user to redo BLOCKSWORLD
  ~clear screen
(WINCLR)
  ~prompt for redo question
(PCENTER 10 80 "TO END PRESS E, TO REDO PRESS R")
  ~clear keyboard buffer
  ~read response and loop to appropriate label
(CLRBFI)
(COND
  ((EQ (TYI) "R")  ~clear screen
                 (WINCLR)
                   ~prompt for redo choices
                 (PCENTER 10 80 "TO START FRESH PRESS F")
                 (PCENTER 12 80 "TO START FROM CURRENT STATE PRESS C")
                   ~clear keyboard buffer
                 (CLRBFI)
                   ~read response and loop accordingly
                 (COND
                   ((EQ (TYI) "C")(GO REDO_GOAL_STATE))
                   (T (GO REDO_INITIAL_STATE)))  )
  ~clear screen and prepare to end
  (WINCLR)
  ~end of function BLOCKSWORLD
]

~***************************************************************************
  ~function STATE_DEF_IN, first student assignment

(DEFUN *STATE_DEF_IN (LAMBDA (STATE_NAME)
(BEEP)
(COND
    ~if STATE_DEF_IN function has been defined execute it
  ((GT (LENGTH (GETD 'STATE_DEF_IN)) 2)(STATE_DEF_IN STATE_NAME))
    ~if STATE_DEF_IN has not been written then give assignment
  (T   ~clear screen
     (WINCLR)
       ~display assignment
     (PCENTER 1 80 "Assignment => STATE_DEF_IN function")
       ~assignment text assigned to variable ASMT_STATE_DEF_IN_TEXT
     (PTEXT 3 80 ASMT_STATE_DEF_IN_TEXT)
  ~end of function *STATE_DEF_IN
]

~***************************************************************************
```

```
  ~function DEPTH_FIRST, second student assignment

(DEFUN *DEPTH_FIRST (LAMBDA (Q)
(COND
    ~if DEPTH_FIRST function has been defined execute it
  ((GT (LENGTH (GETD 'DEPTH_FIRST)) 2)(DEPTH_FIRST Q))
    ~if DEPTH_FIRST function not defined then give assignment
  (T   ~clear screen
     (WINCLR)
       ~display assignment
     (PCENTER 1 80 "Assignment => DEPTH_FIRST, POSSIBLE_MOVES_GEN")
     (PCENTER 2 80 "              NEW_STATE_DEF, DUP_STATE, GOAL?")
     (PTEXT 4 80 ASMT_DEPTH_FIRST_TEXT)
  ~end of function *DEPTH_FIRST
]

~***************************************************************************
  ~third student assignment PRINT_PATH

(DEFUN *PRINT_PATH (LAMBDA (QU)
  ~if PRINT_PATH function has been defined, execute it
(COND
  ((GT (LENGTH (GETD 'PRINT_PATH))2)(PRINT_PATH QU))
      ~if not then give assignment
  (T  ~clear screen
      (WINCLR)
      (PCENTER 1 80 "Assignment => PRINT_PATH")
      (PTEXT 3 80 ASMT_PRINT_PATH_TEXT)
      ~end of function *PRINT_PATH
]

~***************************************************************************
  ~function to center string on given row

(DEFUN PCENTER (LAMBDA (ROW WIDTH STRING)
  ~locate print start position
(WINCURS (LIST ROW (-(/ WIDTH 2)(/ (STRLEN STRING) 2))))
  ~print string to screen
(PRINC STRING)
  ~end of function PCENTER
]


~***************************************************************************
  ~function PTEXT is a rather long utility function to format a
  ~long text string for screen print and is not included in the
  ~appendix due to its size. This dummy function replaces it for
  ~this discussion.?
(DEFUN PTEXT (LAMBDA (A B STR)
(PRINTC STR)]

~***************************************************************************
  ~this function is system dependent and is left to students to design
  ~if a graphic display assignment is desired
(DEFUN SHOW_STATE (LAMBDA (XX)]

~***************************************************************************
  ~initialize student functions to NIL
```

```
(DEFUN STATE_DEF_IN (LAMBDA ()]
(DEFUN DEPTH_FIRST (LAMBDA ()]
(DEFUN PRINT_PATH (LAMBDA ()]

~*********************    ASSIGNMENT  TEXT   ******************************
  ~text string for ASMT_BW_TEXT

(SETQ ASMT_BW_TEXT
"Introduction: Blocksworld  is  an AI program  which has the ability to
               manipulate  its  environment  in  order to realize  some
               user specified goal.  The  BLOCKSWORLD  environment is a
               set of up to ten uniquely labeled cubes  (blocks), which
               are placed in  the' environment  in some user  specified
               INITIAL   configuration.    A  configuration  of  blocks
               consists of up to ten stacks of blocks  defined  in some
               stack order.    Example:  A on B on TABLE,   C on TABLE,
               F on G on D on TABLE, etc.   Given the constraints of up
               to ten unique blocks and up  to ten table locations upon
               which to place the blocks  (or stacks of blocks),  there
               are  335,221,280  possible INITIAL state configurations,
               any  of  which  may  be  selected.  Once the BLOCKSWORLD
               program  is  given  its  INITIAL configuration  (INITIAL
               STATE),  the  user  must specify some new  configuration
               (GOAL STATE), the BLOCKSWORLD program will determine, on
               its  own,  how it can manipulate the INITIAL state  into
               the  GOAL  state.   The  manipulator  in  BLOCKSWORLD is
               defined  to  be a fictitious robot hand with the ability
               to UNDERSTAND four commands.
                 (1)-PICKUP(X) Pick some block off the table,
                 (2)-PUTDOWN(X) Put its held block down on the table,
                 (3)-UNSTACK(X Y) Remove the block X from top of Y,
                 (4)-STACK(X Y) Stack held block X on to block Y.
               With these four moves and the additional constraint that
               only  one block at a time may be moved, BLOCKSWORLD must
               develop a  solution,  (a list of moves required to reach
               the GOAL state).")

~**************************************************************************
  ~text string for ASMT_STATE_DEF_IN

(SETQ ASMT_STATE_DEF_IN_TEXT
"Write a LISP function STATE_DEF_IN with the following attributes:
ARGUMENT -  The single argument is the string of the name of the  state
            for which the function is soliciting information.
FUNCTIONALITY - The  function  must  be  user  interactive,   with  all
            appropriate input  error  checking,  ex:  blocks > 10,  two
            names the same etc.  The prompts must unambiguously  obtain
            all state information and convert this information   into a
            symbolic state definition.
RETURN  -   The state definition in a list form. Use only the following
            state descriptors:
            (ONTABLE  (X)) -> Where X is some literal name of a block.
            (ON (X Y)) -> Where X is on top of Y.
            (CLEAR  (X)) -> Where the block X is clear on top.
            (HAND  EMPTY) -> Which indicates that the hand is empty.
            (HOLDING  (X)) -> Where X is the held block.
Note: Make each state descriptor a list itself and the complete
state definition is a list of these descriptors.")

~****************************************************************************
  ~text string for ASMT_DEPTH_FIRST

(SETQ ASMT_DEPTH_FIRST_TEXT
"Write a LISP  function  named  DEPTH_FIRST  with  the  following
           attributes:
ARGUMENT - The list (queue) whose first elements first element is
           the state definition of the current state.
FUNCTIONALITY -  Perform a depth first search procedure using the
            global  variable GSTATE as the goal state  for  which
            you are searching.  If the state definition  on  the
            front  of the list is not the goal state  then  depth
            first search requires that you mark the queue and add
            the children  (next possible states) to the front  of
            the queue,  if those children (states) have not  been
            visited  before.  In  order  for  you  to  add   the
            children,  you  must  know what they  are;  you  must
            generate the next possible states from the conditions
            which  presently exist, ie. current state,  by  using
            the constraints imposed on the robot.  You will  need
            a function named  POSSIBLE_MOVES_GEN  which  receives
            one argument,  the  current  state  definition  which
            returns  a  list  of  the  possible  moves  from this
            current state.
            (ex. POSSIBLE_MOVES_GEN  (CAR Q)) =>  ( (PICKUP  (A))
            (PICKUP (B)) (UNSTACK (D C)) ).  The elements in this
            list  can  be  applied to the current  state by  some
            production system  (embodied in a  function  you may
            call NEW_STATE_DEF).  These next possible states can
            now be added  to the front of the list (queue) in the
            normal  depth  first  search way  and  recursion  can
            continue  until  the goal state is found.  Note:  To
            conserve memory you need only apply the first of  the
            possible  moves to the current state to get just  one
            next  state  and  maintain on the  list  (queue)  the
            possible moves  for later path information.
RETURN  -   DEPTH_FIRST  must return the list (queue)  containing
           the  state  information  about the path  from  initial
           state to goal state.")

~****************************************************************************
  ~text string for ASMT_PRINT_PATH

(SETQ ASMT_PRINT_PATH_TEXT
"Write a LISP function with the following attributes:
ARGUMENT - The single argument is the list of the path information
           (the queue) returned by the DEPTH_FIRST function.
FUNCTIONALITY -  This function must operate on the list argument and
           print the solution to the search (the path) to the screen.
RETURN - This  function  should  return the path information.")
```

UNIVERSAL LOGIC IMPLEMENTER: A GENERAL PURPOSE TOOL FOR A DIGITAL LOGIC DESIGN LABORATORY

Mark L. Manwaring, Veljko D. Malbasa and David A. Seamans
Department of Electrical and Computer Engineering
Washington State University
Pullman, WA 99164-2752

ABSTRACT

In this paper we present a general purpose digital logic implementer, a tool designed to serve as a workstation for laboratory experiments in the digital logic and computer architecture courses taught in the Electrical and Computer Engineering Department, Washington State University. The logic implementer is a versatile tool suitable for a wide spectrum of experiments, from simple gate-level implementation of logic circuits to the implementation of microprogrammed bit-slice processors. The basic features of the logic implementer include generality of applications, easy-to-use environment and robust, reliable interconnections implemented during the laboratory experiments. As an especially valuable characteristic of the implementer we emphasize the possibility of solderless, rapid prototyping. We discuss the basic parts of the logic implementer, and we describe some experiments for courses taught at different levels.

I. INTRODUCTION

Courses in the digital logic design and computer architecture areas constitute a necessary part of the curriculum in the Electrical and Computer Engineering Department at Washington State University. From the time that these courses were first developed, questions of adequate laboratory workstations have been raised. While software laboratories were organized and started without significant problems by installing a number of terminals and/or personal computers, setting up hardware-oriented laboratories has not been so easy. This problem has been studied extensively by the authors, the main goal being to obtain an advanced, general purpose tool suitable as a student workstation in a number of hardware projects with various degrees of complexity. We wanted to design a laboratory facility where the students could get practical experience appropriate to the level of their study.

One can find in the references a number of approaches which have been taken to solve the problem. The main objectives were to implement a specialized workstation suitable for projects with small scale integrated (SSI) or medium scale integrated (MSI) circuits (for example see [1]), or to implement workstations for hardware projects with microprocessors or bit-slice processors, [2-5]. A general objection to these approaches is that one has to implement a different set of workstations for each particular area, a problem which causes extra expense and which requires a new set of workstations for every new laboratory-oriented course. We tried to overcome this problem by designing a Universal Logic Implementer (ULI). The main goal of the ULI project was to provide a reliable, general purpose, easy-to-use tool which can meet specific experiment requirements by the simple insertion of cards oriented to specific experiments. The student responsibility in the experiment is to design the logic circuit, to choose suitable cards and to make the connections necessary to implement the circuit. We decided the ULI workstation should meet the following requirements:

- Be universal enough to cover a wide spectrum of hardware experiments with different levels of complexity.
- Provide all the necessary logic gates, computer subunits (such as memories, ALU's or controllers) and input/output devices (switches, LED's, keyboards,and possibly floppy-disk drives), leaving the student to design the digital system and interconnect the parts which are provided.
- Be simple to use, and to include the possibility of making the interconnections on a detachable panel remotefrom the workstation. This would enable the students to upgrade their circuits gradually, since they wouldnot have to destroy the connections at the end of each lab period.
- Be compatible with some off-the-shelf cards, which will automatically provide an additional source of cards.

- Be self-contained and provide all necessary parts for a particular experiment.
- Be suitable for further expansion and upgrading.
- Be a rugged, reliable implementation.

We decided to take a modular approach to the implementation of the ULI by dividing a workstation into two parts: the station itself with the hardware units common to all experiments and a set of experiment-oriented cards. This approach enables incremental growth of the ULI workstations, keeping the cost of the whole project low. Also, we believe that, by designing new, state-of-the-art cards as needed, we will have updated ULI workstations in the long term.

Design and construction of the ULIs started in 1985, and the first laboratory use was in fall, 1987 in EE 214, Digital Logic Design. The project was under the direction of the members of the faculty, but its implementation has been mainly by students and staff. The ULI workstations will eventually be used in the laboratories for the following courses:

EE 214	Digital Logic Design
EE 314	Microprocessor Systems
EE 414	Fundamentals of Digital Systems
EE 424	Digital System Architecture
EE 434	VLSI Systems I

There is also a potential for use of the ULI in graduate student projects since the students can build their own cards and then use the ULI workstation to implement the project. Also, in the VLSI Systems course, ULI workstations can serve as a testbed for testing the chips which the students have designed.

In the sections that follow, we first present an overview of the ULI, and then discuss its basic subunits and the set of cards. At the end of the paper we describe some typical experiments to be performed on the ULI.

2. OVERVIEW OF THE UNIVERSAL LOGIC IMPLEMENTER

We can recognize two main parts of the ULI workstation: the ULI station itself and a set of cards. The ULI station is application-independent, and provides all hardware units necessary to implement any experiment: input and output devices, power supply, connectors, etc. The set of cards, each containing a particular set of digital gates or computer hardware subunits, represents the experiment-dependent part of the workstation. Choosing different card subsets, one can make a number of different ULI configurations suitable for a wide spectrum of digital hardware projects.

The basic block diagram of a ULI station is shown in Figure 1.

Physically, we partition the ULI station into two parts: the control panel (CP) and patch panel (PP). The CP is the experiment independent part, and it provides all the necessary input and output devices. The CP thus provides a useful interface for controlling and observing the digital circuit which the student has built.

The PP has 6 card slots which are wired to patch connectors. By means of patch connectors and a set of wires with connectors to match patch panel connectors, one can make connections between the inserted cards, thus interconnecting card circuitry. The wire connectors are stacking connectors, thus allowing one to make several connections from the same patch hole. There is a similar patch connector on the CP which facilitates the connection between the implemented logic circuit on the PP and input/output devices on the CP. The process of making the connections between the patch connectors we will call "patching."

There are two types of patching: between the cards themselves and between the cards and CP. The former patching is used to implement the digital circuit; the latter is used to connect the input/output devices to the circuit.

The student should take the following steps during the experiment procedure:

1. Analyze the problem and design the digital system.
2. Choose from those available the subset of cards necessary to implement the digital system.
3. Make a patching list of all connections between the cards and between the digital system and the CP.
4. Do the patching and test the digital system.

To implement a designed digital system on the ULI, one should first draw a standard schematic diagram similar to those which are used, for example, in wirewrapping. Thus, the experience gained by working with the ULI will be valuable when a student has to use another method of digital circuit implementation. The advantage of using the ULI is that it allows rapid assembly of a digital system.

A photograph of the ULI with two patch panels is shown in Figure 2.

On the front side of the PP there are 6 patch connectors which are connected with 6 S-100 compatible connectors on the back side of the PP. The S-100 compatibility makes it possible to include S-100 off-the-shelf cards in the set of experiment- dependent cards. The CP patch connector is used to access the CP input/output (I/O) devices from the patch panel.

It should be pointed out that the PP is removable and that we plan to have several PPs with each station. This will allow the use of the same station by several students. Each student will make connections on his PP and will use the station only to test the implemented digital circuit, which may take place over a period of several weeks.

3. CONTROL PANEL

The basic motivation for implementing the CP was to provide the student with easy-to-use devices which are common to most of the experiments. The CP incorporates:

- -input devices:
    - 16 three-position switches
    - 4 hexadecimal thumbwheel switches
    - hexadecimal keypad
    - clock generator
    - clock frequency selector switch
    - pushbutton for step-by-step clock pulse generation
- -output devices:
    - 32 two-color LEDs
    - 4 seven-segment displays
    - voltmeter
- -RS-232 connector
- -patch connector
- -power supply unit

Figure 3 shows the front view of the CP.

All input and output signals of the CP are available on the 100-pin CP patch connector. These signals, together with the built-in I/O devices of the CP, are controlled by the control logic of the CP. The input and output signals of the control logic are shown in Figure 4.

The control logic may be defined as performing the function f in the equation

$$Y = f(X)$$

where X is the set of input signals and Y is the set of output signals. The function f is implemented by an 8085-based microcomputer. This approach offers relatively simple implementation of the control logic, and offers the possibility of changing the function f. We look at the microcomputer as a controller with a default control function f stored in ROM, but with the possibility of accessing the controller via the RS-232 line to change the function f. This option greatly enhances the generality of the ULI, making it possible to change the logic connections between the input and output signals.

Although the RS-232 line is designed primarily for modification of the controller program, it has another use also. One of our long term goals is to build software tools for design and simulation of digital circuits and to use them together with the ULI. For example, one can design and simulate a digital circuit on a general purpose digital computer. The output signals which the simulated circuit will produce for some input signals can be stored in the computer memory. Then the same circuit can be implemented on the ULI and it can be stimulated by the same input signals. Via the RS-232 line the output signals of the real circuit can be transferred to the computer and compared with the simulation results. Some software tools, like the BOOZER program for minimization of the Boolean functions, (see [6]), are already available to the students.

To increase the usefulness of the CP, we implemented the interface circuit for the CP patch connector in a way that allows declaring particular patch holes either as input or as output holes. This makes the whole configuration very flexible so that the student can use any device from the CP with little effort.

4. THE CARDS

The main purpose of the set of cards is to provide the basic building elements for the experiments. As the ULI stations are designated to be used for sophomore, junior and senior level projects, the choice of cards is closely related to the laboratory projects for these courses. We will classify all the cards into 3 groups: generic, dedicated, and off-the-shelf.

Generic Cards

These cards are designed to provide access to all signal pins of the integrated circuits which are installed on them. There is no connection between any of the pins of the integrated circuits on the board, all connections being made by the students on the PP. We decided to implement these cards with sockets and then to insert various integrated circuits in the sockets as needed. There are 5 different configurations of the generic cards:

- Type 1: cards with six 16-pin sockets
- Type 2: cards with two 28-pin and two 16-pin sockets
- Type 3: cards with four 20-pin sockets
- Type 4: cards with a 40-pin and 28-pin socket
- Type 5: cards with a proto board installed

Examples of integrated circuits that will be installed in the cards are shown in Table 1. Obviously, once you install the integrated circuits, the generic card will become a dedicated card.

Dedicated Cards

These cards are to provide some basic digital subsystems for more complex digital systems. Typically, they are designed with several integrated circuits and with all the necessary interconnections to implement a particular subsystem. We are initially designing two types of dedicated cards: programmable controller and processor unit.

The programmable controller, shown in Figure 5, is based on the TI 74AS890 microsequencer unit [7].

Although with the 74AS890 microsequencer it is possible to implement a high performance programmable controller, we decided to give the priority to the classic approach, leaving as much freedom as possible to the student to define a particular configuration of the controller. Consistent with this approach, we did not assign a meaning to all the input and output signals, and we will not suggest the application of the controller together based with an ALU based on a 74AS888 in assembling a central processing unit. Instead, the controller design enables its use as a general purpose digital system controller.This means that it is completely up to the experimenter to choose the controller for a particular design project, to assign meanings to particular signals, and then to write the microcode.

The card with two independent processor units, shown in Figure 6, is based on an 8-bit slice processor, the TI 74AS888, [7].

All input and output signals are available at the patch connector, which gives the experimenter the option of using one 8-bit processor, of making up a 16-bit processor by using both processors, or of implementing a 32-bit processor with two cards.

Off-the-Shelf Cards

The connectors on the back of the PP are S-100 compatible except that we use a 5V supply voltage instead of 8V. This makes it possible to include S-100 off-the-shelf cards in the set of cards for the ULI. For example, since our microprocessor laboratory uses 68000 microprocessors and HP 64000 development system workstations, we decided to include 68000-based CPU, memory and I/O. These cards will be used for projects in the microprocessor course so that the students can implement a circuit on the ULI and use an HP 64000 to analyze the circuit and write the programs.

5. EXPERIMENTS

In this section we will discuss briefly three different experiments which use or will use the ULI. The experiments are selected to demonstrate the way the ULI can be used in projects of different levels of complexity. The first experiment is for a sophomore-level introductory class on digital design, the second for a senior-level class on digital system architecture, and the third one represents a proposal for an advanced-level experiment which can be organized using the ULI.

Sophomore Level Experiment

At this level the students learn the basics of digital logic design, including how to use gates and basic memory elements, and how to implement simple combinational and sequential circuits. We present the following problem as a typical introductory level lab assignment.

Suppose a secret room has a secure door with three different locks, the keys of the locks being with three different persons A, B and C. The door will be unlocked if at least two people insert their keys. If the events of inserting the keys are simulated by closing three switches on the control panel and if the event of unlocking the door is simulated by turning on a light emitting diode, find the minimal Boolean function which describes the condition of unlocking the door, and implement it.

The truth table, K-map, resulting function and the two-input NAND gate implementation are shown in Figure 7.

Now the student must choose a card with appropriate gates. There is a laboratory data book with all necessary information about the cards, PP and CP. In our example the student will choose the card with NAND gates. Part of a laboratory data book page with the card pin assignments of the two gate packages is shown in Figure 8.

The next step is to choose one out of six patch connectors on the PP, denoted as A, B, C, D, E and F. If the student chooses connector A, then the pin assignments of the card will be modified by adding a prefix A. For example pin B5 becomes A-B5 which means:

A: the number of the row on the patchboard connector
B: the number of the column on the patchboard connector
5: patchboard connector A

The next step is to fill out the patching list on the basis of the circuit from Figure 7 and the pin assignments from Figure 8. The resulting list for this example is shown in Table 2 and in Figure 7. The control panel patch connector is denoted by P.

The student will now patch the wires according to the entries of the first two columns of Table 2. Having finished the patching, the student will check each connection and put the marks in the "check" column. The final step is to test the circuit by closing the input switches and observing the output LED. When the input conditions are met the output LED will be turned on, if the design and implementation processes were performed correctly.

The ULIs were first put into use in the fall semester of 1987 in the sophomore digital logic design course, EE 214. They proved to be a practical and flexible means of implementing the laboratory work in the course as we had hoped. The students and teaching assistants were able to learn the new connection procedures quickly, and the students were soon able to carry out the assigned laboratory projects successfully. One slight disadvantage of this system reported by the students is a feeling of remoteness from the components being connected in the circuits. This feeling can be overcome, however, we believe, by assuring that the students take time to examine the component cards at the beginning of the laboratory period, and plug the cards into the sockets themselves.

Senior Level Experiment

At this level the students will be familiar with the concept of the programmable controllers and with the partition of a digital system into the control and data path units. Because of the complexity of the experiments at this level, we will not give all the details of the experiment, assuming that a procedure similar to the one explained in the example of the sophomore level experiment will be applied.

A typical, three week lab assignment at the senior level will be to assemble an individually designed 16-bit computer system with:

- memory unit
- ALU
- set of 16, 16-bit registers
- specified set of status bits
- microprogrammable controller
- one input and one output register

The experiment includes the writing of the microcode to implement a specified list of instructions. The microcode is to be written into the microcode store, and the entire system must have all its operations tested.

In this experiment the student will use four cards:

- the programmable controller card
- the data path units card
- the type 1 generic card with four 4-bit registers for input/output operations and a decoder for the select operation
- the type 3 generic card with ROM and RAM for use as a memory unit for testing

During the design stage the student will:

- design the logic diagram of the computer system
- design the detailed logic diagram of the CPU
- design the connections between the parts of the system
- design the simple input/output subsystem with two registers

At the end of this stage the student will become familiar with the processor and the controller, and will be able to write the microcode and a simple monitor program. The implementation phase includes:

- patching the units according to a table similar to Table 1
- writing the microprogram into the microstore
- writing the simple monitor program into ROM
- testing the basic function of the system.

During the implementation phase, the test procedure will be quite complex. Therefore, it will be suggested that the students test particular units separately, and then test the overall system. Testing of the units can be performed either by means of a logic state analyzer or by simulating input data and control signals of a unit and selecting the step position of the clock selector. All the output signals can be observed on the LEDs of the control panel.

Advanced Level Experiment

The advanced level experiments are not yet prepared, but implementing them is the ultimate goal of the ULI. To demonstrate the power of the ULI as a laboratory work station, we will briefly describe a proposed advanced level experiment. This one can take several weeks and, if carefully planned, the laboratory work for the whole semester can consist of this single project.

In this experiment the students will start with the experiment described as a senior level example. The next hardware steps would be to include interrupt and direct memory access (DMA) facilities. By using an RS-232 interface it is possible to connect a terminal to the implemented computer system.

Simultaneously, the students can start writing pieces of the software, beginning with the monitor program and a handler for the terminal. Connecting a floppy disk drive would add a new dimension to the project, making it possible to write some of the routines of an operating sytem.

The experiment described will give to the students a thorough insight into the architecture of digital computers. They will experience all the phases of computer design, starting with implementing the ALU, controller and the other subunits of the computer, then defining the microcode, interfacing the peripheral devices, and ending with generating the basic routines of an operating system.

6. CONCLUSION

The universal logic implementer system has been tested in student laboratories through the 1987-88 academic year, and it has been demonstrated to be a powerful tool for laboratory experiments. With an appropriate library of cards with a wide range of functions available, it is expected that it will meet the needs of laboratory experimentation and prototying throughout the curriculum of digital hardware courses. Further, it has application to some graduate-level research experiments. At the graduate level, student's work may involve the design of an integrated circuit or a combined function of available integrated circuits. The ULI provides an implementation medium where the students can rapidly assemble, debug, and modify a whole, complex system. The key feature is that it allows rapid assembly of complex circuits while retaining the advantage of component reusability by other users. The authors feel that the ULI is a significant advancement in the implementation of digital hardware circuits for laboratory and experimental environments over contemporary methods.

7. ACKNOWLEDGMENTS

The authors wish to thank the students and staff who took part in the realization of the ULI project: William Baker, Michael Dierken, Rodney Johnson, Clifford Fallon, Timothy Heldt, Rick Hoover, John Hunter, Steven Savage, and Carl Wells.

Also we wish to acknowledge our department for supporting the concept and the implementation of the project, and INTEL Corporation for donating the majority of the logic components for the control panel.

REFERENCES

[1] G. G. Langdon and D. W. Lewis, "An approach to a modest digital laboratory adjunct for a logic course," *IEEE Trans. Educ.*, vol. E-21, pp. 35-36, Feb. 1978.

[2] F. J. Looft and W. H. Eggimann, "Microprocessor-based projects at Worcester Polytechnic Institute," *IEEE Trans. Educ.*, vol. E-26, pp. 85-91, Aug. 1983.

[3] C. W. Merriam, "Computer hardware design projects for undergraduate students," *IEEE Trans. Educ.*, vol. E-26, pp. 77-85, Aug. 1983.

[4] D. L. Dietmeyer and C. R. Kime, "A bit-slice computer design laboratory," *IEEE Trans Educ.*, vol. E-24, pp. 113-119, May 1986.

[5] C. H. Roth and G. Minassian, "A bit-slice microprocessor system for teaching microprogramming," *IEEE Trans. Educ.*, vol. E-24, pp. 119-122, May 1981.

[6] W. I. Fletcher, *An Engineering Approach to Digital Design*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[7] *SN74AS888, SN74AS890 Bit-Slice Processor User's Guide*. Texas Instruments, 1985.

TABLE I
Integrated Circuits to be Installed on Cards

Card Type	Integrated Circuits
1	NOR or/and NAND Gates, Multiplexers, Ddecoders, Flip-flops, Counters
2	ROMs (or EPROMs), Decoders
3	Bus Buffers
4	Microprocessor, ROM
5	Other Devices

TABLE II
Example of Patching List

From	To	Mnemonic	Check	Comment
P - A5	A - A2	A(H)	x	Input A,B, and C from
P - B5	A - E2	B(H)	x	the three switches of
P - C5	A - D3	C(H)	x	CP
P - A8	A - D5	Y(H)	x	LED
A - A2	A - E3	A(H)	x	input A (H)
A - E2	A - B2	B(H)	x	input B (H)
A - D3	A - A3	C(H)	x	input C (H)
A - C2	A - E5		x	(AB) (L)
A - D2	A - D4		x	(BC) (L)
A - A4	A - C4		x	(AC) (L)
A - B4	A - A5		x	(BC + AC) (H)
A - B4	A - B5		x	(BC + AC) (H)
A - C5	A - A6		x	(BC + AC) (L)

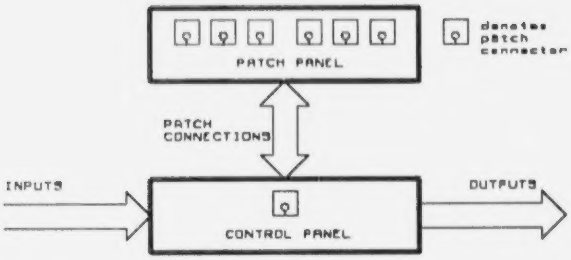

PATCH PANEL
denotes
patch
connector
PATCH
CONNECTIONS
INPUTS
OUTPUTS
CONTROL PANEL

Figure 1. Block diagram of ULI station.

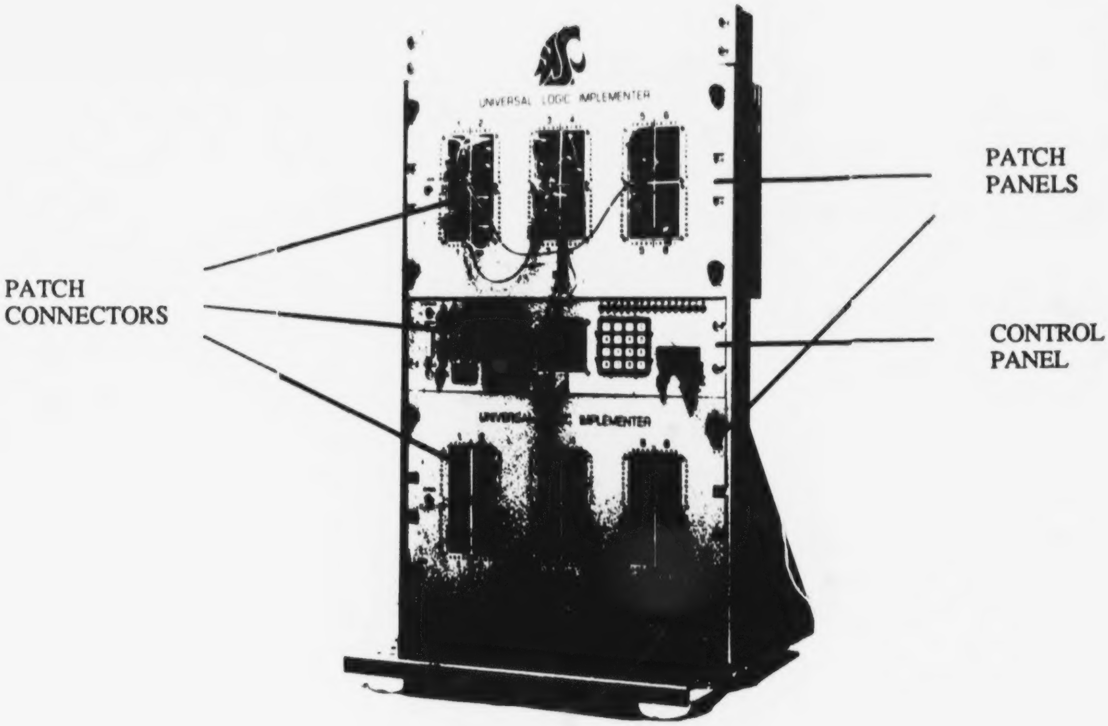

UNIVERSAL LOGIC IMPLEMENTER
PATCH
PANELS
PATCH
CONNECTORS
CONTROL
PANEL
UNIVERSAL IMPLEMENTER

Figure 2. Photograph of ULI.

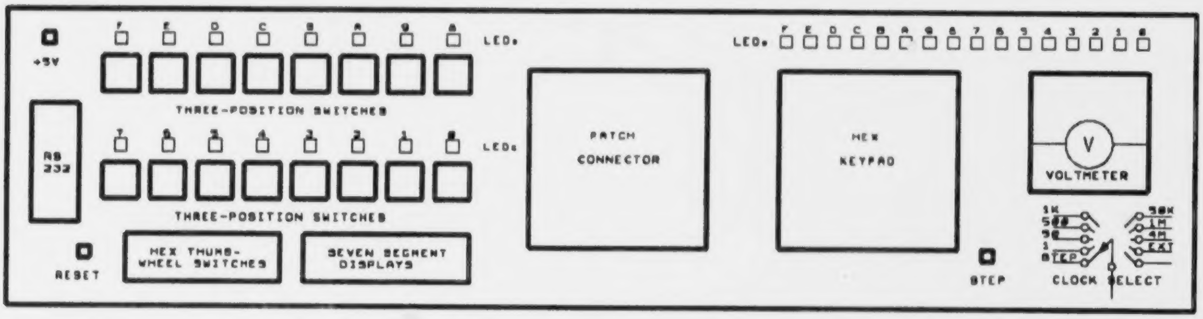

+5V
F
E
D
C
B
A
9
8
LEDs
THREE-POSITION SWITCHES
RS
232
7
6
5
4
3
2
1
0
LEDs
THREE-POSITION SWITCHES
RESET
HEX THUMB-
WHEEL SWITCHES
SEVEN SEGMENT
DISPLAYS
PATCH
CONNECTOR
LEDs F E D C B A 9 8 7 6 5 4 3 2 1 0
HEX
KEYPAD
V
VOLTMETER
STEP
1K
500
50
1
STEP
50K
1M
4M
EXT
CLOCK SELECT

Figure 3. Front view of the control panel.

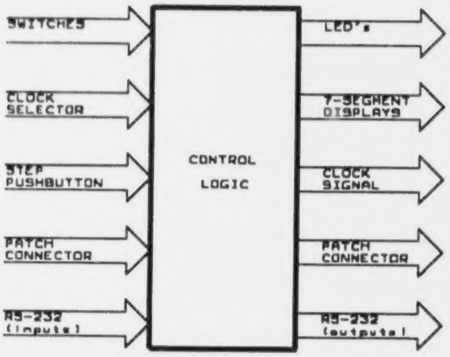

SWITCHES
CLOCK
SELECTOR
STEP
PUSHBUTTON
PATCH
CONNECTOR
RS-232
(inputs)
CONTROL
LOGIC
LED's
7-SEGMENT
DISPLAYS
CLOCK
SIGNAL
PATCH
CONNECTOR
RS-232
(outputs)

Figure 4. Input and output signals of the control logic.

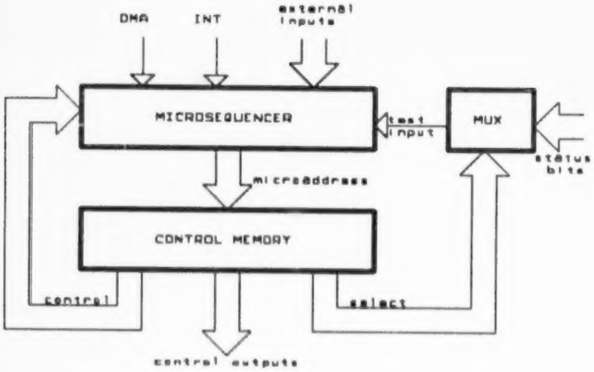

DMA
INT
external inputs
MICROSEQUENCER
test input
MUX
status bits
microaddress
CONTROL MEMORY
control
select
control outputs

Figure 5. Block diagram of programmable controller.

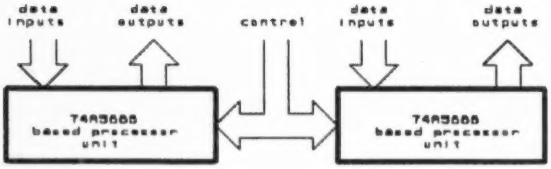

data inputs
data outputs
control
data inputs
data outputs
74AS888 based processor unit
74AS888 based processor unit

Figure 6. Block diagram of processor unit card.

A(H)	B(H)	C(H)	Y(H)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$Y(H) = (AB + BC + AC)\ (H)$$

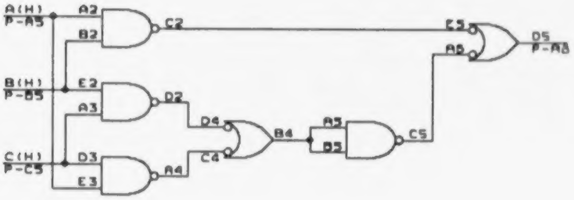

A(H)
P-A5
A2
B2
C2
B(H)
P-B5
E2
A3
D2
C(H)
P-C5
D3
E3
A4
D4
C4
B4
A5
B5
C5
E5
A6
D5
P-A8

Figure 7. Typical sophomore-level logic design.

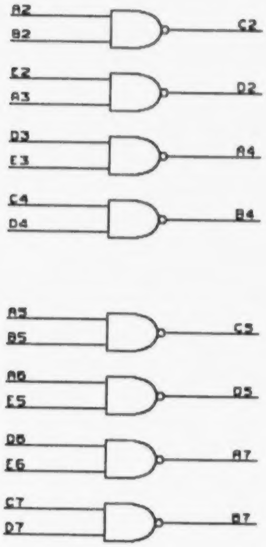

A2
B2
C2
E2
A3
D2
D3
E3
A4
C4
D4
B4
A5
B5
C5
A6
E5
D5
D6
E6
A7
C7
D7
B7

Figure 8. Typical logic card diagram.

THE SEARCH FOR A SUITABLE ACADEMIC MODEL FOR COOPERATIVE EDUCATION SUPERVISION IN COMPUTER SCIENCE

Elizabeth S. Adams
The American University
Department of Computer Science & Information Systems
206 Hurst Hall
Washington, DC 20016
202-885-3141 (desk) 202-885-1470 (messages)
bgoldin@auvm (bitnet)

ABSTRACT

The difficulty facing a faculty member assigned to supervise computer science majors enrolled in cooperative education is that of ensuring that the learning experience of the student is academically equivalent to that of a standard course. Unsatisfactory alternatives that have been experimented with are outlined and problems making it difficult to find an ideal solution are discussed. At the heart of this paper is the presentation of a model for providing a legitimate academic framework for a computer science cooperative education experience. The model, which is described in some detail consists of: class meetings with related assignments; library research culminating in a paper; oral presentations; group discussion of work enviornments and problems; site visits by the co-op faculty supervisor; pre and post placement sessions at the cooperative education office and software demonstrations. The grading structure is provided along with an explanation of the motivations behind it. The appropriateness and flexibility of the model are discussed along with student attitudes towards it.

PART 1: INTRODUCTION

A survey of the cooperative education literature indicates that a lot of attention has been paid to the following issues: program establishment; employer recruitment; cooperative education as a feature in attracting students to an institution; means of attaining and retaining external and internal support for cooperative education; whether the cooperative education office should be included in or separate from the career development office and what the relationship between them should be; whether co-op supervisors should be faculty members; the recruitment and training of faculty members and others who supervise co-op; how those who supervise co-op should be compensated; whether students should receive no credit, additive credit, or regular degree credit for their co-op participation; whether co-op should be required or optional; whether co-op placements must be major-related; whether co-op should occur in parallel with other courses or separate from them. Little attention seems to have been paid to the issue of what the academic component of co-op ought to be. This paper presents a model for the academic component of a computer science cooperative education placement.

Cooperative education was brought to life in 1906 by Dean Herman Schneider of the University of Cincinnati. Eighty years later, 101,829 students were enrolled in cooperative education programs in the United States and Canada [2].

At The American University students can earn degree credit for cooperative education. They may apply no more than eighteen (18) credits of cooperative education towards an undergraduate degree amd no more than six (6) towards a graduate degree. The maximum amount of credit per placement is set by the University at nine (9) credits but the actual credits per placement may depend heavily upon the policies of the department which has approved the position and whose faculty member will supervise the placement. Co-op credit may not be used to fulfill major requirements in Computer Science although rare exceptions may be made by the faculty supervisor acting with the approval of the department chair based upon the nature of the job and the student's background. The general policy in computer science is to allow no more than three (3) credits per placement amd to actively discourage more than two placements. By University policy, once having earned co-op credits for a placement, students can not continue in the same job for additional credits. Participation in the cooperative education program is optional. Students may register for co-op during the regular semester, in parallel with other courses -- in which case their job placement is usually part-time. Alternatively they can work in a full time placement, either in the summer or during the school year taking few or no parallel credits.

PART 2: STUDENT VIEWPOINTS

My students' motivations for participating in cooperative education appear to be threefold: to get some "somewhat" relevant experience to put on a resume to make it easier for them to get a fulltime job after graduation; to earn some money, either to pay for some portion of tuition or for expenses; and to escape from the rigor/formality/structure of ordinary college courses. Most of them see it as "less demanding" than a course -- even if a greater number of hours are spent on it. Most of my students would prefer not to have to do anything beyond the tasks at work to receive their grades and academic credit. They would also like to receive major credit rather than elective credit. While they are reluctant to pay for more than three (3) credits, the more credits they can earn, the better they like it.

PART 3: MY VIEWPOINT

I am in the position of assigning a grade which represents my evaluation of what a student has learned. A passing grade will earn students credit towards the degree equivalent to that earned for standard university courses. While I could make the grade entirely dependent upon their work supervisor's evaluation of on the job achievement and performance, since at a bare minimum they are obviously learning a great deal about functioning in the real world, I am not comfortable with that. It is my belief that a formal demonstration of learning in the form of a paper is a necessity.

PART 4: THE SEARCH

Over the past seven years, I have been searching for a way to provide a work related academic experience for my students on an individualized basis in a way that satisfied them and me. I met with them in my office. We discussed their jobs. On the advice of my colleagues in other departments, I had them keep journals of their experiences and activities on the job. I visited them in their work environments and talked with their bosses. Together, each student and I searched for a topic about which he/she could write a paper that would be both job related and major related. Material students turned in was read, corrected and evaluated by me and sometimes re-done by them. In the end students received a grade which I based on a synthesis of their boss' evaluation and mine. I was never satisfied with the model because I was never sure that the evaluations I was making were consistent. In addition, I could not decide what to do about students whose on the job performance was rated by their bosses as outstanding but whose performance for me was negligible; or those who did excellent work for me but did not perform well or "fit in" well on the job. Sometimes, in spite of admonitions and policies about checking in, students dropped out of my sight. Then I would call them and encourage them to keep to the minimal structure that had been provided. Sometimes they did and sometimes they did not.

PART 5: THE PROBLEMS

At the American University we do not place a large number of computer science students in coop positions. Due to a number of factors not to be discussed here, we do not have an agreement with any public or private agency to provide students to work for them over a continuing period of time. We are in a city of government agencies and government contractors who can only hire citizens. Thus our international students often must self-develop jobs in order to participate in the cooperative education program. This involves finding a job and having the employer make formal arrangements with the University to participate in the cooperative education program. Jobs are very varied in the degree of difficulty and in the level of technical skill required. Some jobs are intellectually stimulating and provide for significant growth in technical knowledge and expertise. Others are less challenging and only provide the opportunity for students to practise what they already know. Still others turn out to be monotonous, menial, essentially mere busy work. Because of the variety of job type, job quality and employers within a given semester, and from semester to semester, it is impossible to know enough about each job and student to create an academic component for each student tailored to their individual needs, background and job environment. Often faculty members do not know which students they will be supervising or what jobs students will hold until after the semester begins. It is also true that without a peer group a student has no measuring stick against which to measure his/her progress and no one to commiserate with about the difficulty of assignments or the number of them. I found myself having to convince students that having to produce a paper for academic credit and for a grade was not unreasonable. They were sometimes resentful at having to produce work in addition to that performed on the job in order

PART 6: THE SOLUTION

During this past year, I have finally developed a model that I and my students feel is a good one. Its workability does not depend on the quality of the job or even on the background of the individual students. This turns out to be an extremely important element in its success. We meet as a group four times during the semester on Sunday afternoons because that does not conflict with anyone's class or work schedule. We share information all semester. They tell each other about their experiences, environments, problems, successes, and goals. I provide backup and structure for the discussions. We discuss everything from speech and clothing to interpersonal relations to technical information. We use the workbook written by Barbara Taylor and Sue Murray [10] as background material for evaluating the work environment and the individual's place in it.

Students receive the syllabus for the course which contains a list of the course requirements and associated percentages (Appendix I) when they register for co-op. This occurs after they have found a position. Thus they know in advance what the requirements for the course are. Included in the syllabus is the assignment for the first of the four sessions. This first assignment consists of two short reports of one to two pages each. In the first, they are asked to discuss their personal reasons for participating in cooperative education. In the second they are asked to write about different aspects of their first day at work. At the first session, we also discuss the assignments for the rest of the course and get to know one another.

In the interval between the first and second meetings of the course each student selects a research paper topic in consultation with me and prepares a preliminary bibliography for the paper.

At the second meeting each student gives a short, but formal, oral presentation to the group on the proposed paper using visual materials in the form of charts or transparencies. After each presentation there is a group discussion of the topic; of the proposed structure of the paper; of what additional information should be included; of what should be omitted; and finally of the presentation itself. Suggestions are made for improvements in the presentation and we all contribute information about additional sources. Each student turns in a written proposal at this meeting for the software that he/she wants to demonstrate at the next (third) meeting. At this second meeting, we also discuss any problems that may have arisen in the work environment and help resolve

At the second meeting students are given a reading list (Appendix II) which contains non-textbook references to "classics" in computer science: some journal articles, some chapters from books, some articles from the popular literature. The list contains some of those references that everyone talks about but students never have time to read. The assignment asks the students to select five of the papers from the reading list and to prepare a one page written summary of each article to be handed in at the third session. They are also asked to indicate on the summary sheet whether the article should be retained on the reading list for the next semester.

At the third session we have software demonstrations where students share their knowledge. A student from Saudi Arabia may show a word processing program that will print text in Arabic; one from Korea, may demonstrate one that will print text in Korean. A student who has had experience with software used to create lessons for computer aided instruction will demonstrate how it can be used. Another will demonstrate the differences between the Pascal compiler on our IBM mainframe and Turbo Pascal (used on our IBM PCs). Still another will demonstrate the use of a communications program in conjunction with a modem to upload and download files between a mainframe and a micro. These demonstrations vary from semester to semester based on the backgrounds and interests of the students. This semester students demonstrated Turbo Prolog, dBASE III Plus, Harvard Presentation Graphics, Smart Software, Lotus 1-2-3, WordPerfect, Storyboard, and Ada using the ADAVANTAGE compiler by Meridan.

The final meeting is the one at which the students present a summary of their papers to the group in oral form. Some semesters, they also provide written copies to their classmates. The oral presentation takes about 10 to 15 minutes per student, depending on class size, and is followed by a question and answer period.

In addition to the group meetings, students are encouraged to stop by my office to report on progress and discuss individual problems, related to either the academic or work-related components of the course. I also make a prearranged site visit to each student during which they are expected to make an informal presentation to me on what it is they are actually doing. I am interested in their ability to explain what they are doing in detail and in perspective and they are urged to spend time thinking about and preparing for this visit.

Two other assignments are required of the student. The first is a one to two page written critique of the course: what they liked, what they didn't like, what they felt was most valuable, what they felt was a waste of their time. They get full credit simply for turning this critique in. For the second, they are shown how to use Storyboard and asked to create an image on disk using it. The subject of the image is anything relating to the course. They have access to all previously created images. Using the software to create the image earns them full credit. They do not have to have created something particularly clever or beautiful.

The final grade is based on all of the semester's activities. Attendance at and participation in the class meetings contributes to the student's grade as does each assignment, written and oral: the paper, the library readings, the site visit, the reports. Oral presentations are intended mainly to provide students with the opportunity to get up and speak in front of others. Thus they are not graded on the quality of their oral presentation -- this is a topic for constructive group discussion. Merely making an oral presentation which follows the topical guidelines earns full credit. The paper is returned to the student with comments and students are expected to make corrections and elucidations as indicated. Other components of the grade are the employer's evaluation and attendance at a pre and post work seminar at the cooperative education office.

Feedback from students has been very positive. They like the group meetings. They like to share their knowledge with others. They like to contribute to a determination of what they will learn. They find the support group very useful, especially when they run into difficult situations at work or ones that they have trouble interpreting in an impartial way. I like the flexibility of being able emphasize a particular area within computer science any time the composition of the student group indicates a need or desire for it. Both the students and I like knowing at the start of the semester just how much each activity will contribute to the total grade. Following this model provides the students with a legitimate academic experience while at the same time attempting to maximize their experiential learning.

BIBLIOGRAPHY

1. *Cooperative Education Employers in the United States*, National Commission for Cooperative Education, Second Edition, 1986.

2. *Cooperative Education in the United States and Canada*, Cooperative Education Research Center, Northeastern University, Boston, MA, 1986.

3. Goldstein, Michael B., NSIEE General Counsel, Policy Issues in Experiential Education, Paper #3, NSIEE, 1982.

4. Gotlieb, Peter M., Dominant Issues in Cooperative Education, *The Journal of Cooperative Education*, Volume XVIII, Number 2, Winter 1981-1982, 21-30.

5. Gring, David M., and Prausnitz, Walther G., Field-Based Experimental Education: Comments on Integration into the Academic Process, *The Journal of Cooperative Education*, Volume XV, Number 3, Spring 1979, 14-21.

6. Harley, Maurice, American Leaders in Cooperative Education: An Oral History, *The Journal of Cooperative Education*, Volume XVI, Number 1, Fall 1979, 5-16.

7. Heinemann, Harry N., Towards a Pedagogy for Cooperative Education, *The Journal of Cooperative Education*, Volume XIX, Number 2, Winter 1983, 14-26.

8. Hyink, Bernard L., Is Cooperative Education, Education?, *The Journal of Cooperative Education*, Volume XIII, Number 2, May 1977, 5-11.

9. Millonzi, Joel, Liberal Education and Cooperative Education - Are They Compatible?, *The Journal of Cooperative Education*, Volume XV, Number 3, Spring 1979, 5-13.

10. Murray, Sue and Taylor, Barbara, *Going-To-Work Book*, Kendall/Hunt Publishing Company,Dubuque, Iowa, 1987.

11. van der Vorm, Patricia T., Jones, Nancy R., and Ferren, Ann S., A Close Encounter of the Co-op Kind, *The Journal of Cooperative Education*, Volume XV, Number

3, Spring 1979, 22-29.

12. Wayman, Valerie S., CO-OP 85 Survey Results: The Cooperative Education Program of The American University, The American University, internal report.

Appendix I
Co-op Course Requirements and Associated Grade Percentages

4%	Four Sunday meetings from 2 pm to 4 pm. Assignments for each are described breifly below. Embellished descriptions will be provided at each meeting for subsequent meetings.
3%	Attendance at pre-work seminar at Career Center
3%	Attendance at post-work seminar at Career Center
3%	Timely completion of Cooperative Education Workbook to be discussed at first meeting
7%	Assignment for first meeting (September 20th) detailed on attached sheet
5%	Presentation of paper proposal in written and oral form (due at second meeting - to be discussed further at at initial meeting)
10%	Library assignment due at third meeting (to be assigned not later than second meeting)
20%	Major paper on a technical topic related to your job in some fashion. A draft of your paper is to be turned in for critiquing on November 20th. It will be returned with comments and suggested corrections and additions by December 1. At last meeting the final version is due for presentation in both written and oral form. The oral presentation should be a summary of the paper
30%	Evaluation of on the job performance by employer
5%	Satisfactory presentation/demonstration at site visit which will be made during the second third of the semester
5%	Software demonstration for your classmates
3%	Written report (one or two pages) on how the co-op experience measured up to your expectations. You shoud address yourself to the work component, the social component, and the academic component. Due on the last day of classes (December 9).
2%	Storyboard "slide"

Appendix II
List of references for Library Assignment - Fall 1987

Brooks, Jr., Frederick P., *The Mythical Man-Month*, Addison-Wesley Publishing Company, Reading, Massachusetts, (1975)

Dijkstra, E.W., GO TO statement considered harmful, *Communications of the ACM*, 11, (1968), 147-148.

Gassman, H.P., The Influence of George Orwell's 1984 on privacy protection legislation, *Comput. Networks* ISDN Syst. 9, Special Issue: So This is 1984, (Jan. 1985), 29-31.

Miller, G.A., The magical nmber seven, plus or minus two: Some limits on our capacity for processing information, *Psychological Review*, 63, (1956), 81-97.

Rosen, S., Programming Systems and Languages 1965-1975, *Communications of the ACM*, 15, 7 (July 1972), 591-600.

Sammet, Jean E., Programming Languages: History and Future, *Communications of the ACM*, 15, 7, (July 1972), 601-610.

Sammet, Jean E., Why Ada is not just another programming language, *Proceedings of the 1983 ACM annual conference*, ACM, New York 1983.

Tracz, W., Confessions of a Used Program Salesman - The Loves in My Life, *ACM SIGPLAN Notices*, 21, 5, (May 1986), 79-81.

Weinberg, G.M., *The Psychology of Computer Programming*, Van Nostrand Reinhold, New York, (1971).

Weinberg, G.M., D.P. Geller and T.W.S. Plum, IF-THEN-ELSE considered harmful, *ACM SIGPLAN Notices*, 10, 8, (August 1975), 34-44.

Weizenbaum, J., ELIZA -- A Computer program for the study of natural language communication between man and machine, *Communications of the ACM*, 9, 1, (January 1966), 36-45.

Weizenbaum, J., *Computer Power and Human Reason*, W.H. Freeman and Company, San Francisco, California, (1976).

Wulf, W.A., and M. Shaw, Global variables considered harmful, *ACM SIGPLAN Notices*, 8,2, (February 1973), 28-34.

UNIVERSITY COMPUTER EDUCATION

Louise Hewitt, Charles Bauer
David Olson
Marilyn Persson, Daniel Canas
Colleen Kendall-Piel

ABSTRACT: An Integrated Approach to University Freshmen Curriculum

Louise C. Hewitt, Departmental HAWK Coordinator, Charles R. Bauer, Associate Professor, Computer Science Department, Illinois Institute of Technology, Chicago, IL

Illinois Institute of Technology offers undergraduate programs in engineering, sciences and letters, architecture, planning and design, and business administration. All freshmen are participants in the university's Project HAWK program, regardless of individual program of study.

The Project HAWK program for freshmen serves four major objectives:

1. Skill development in the use of personal computer productivity tools;
2. Introduction to regular faculty and upperclassmen in the student's area of concentration;
3. Exposure to major areas within the student's discipline via case studies;
4. Enhancement of leadership, cooperation and study skills.

Fourteen to sixteen two-hour class meetings are lead by regular faculty and upperclassmen teaching assistants within the student's major area. The first hour includes self-paced instructional materials on wordprocessing, spreadsheet, database, graphing and communication applications. These provide the knowledge base of productivity tools for HAWK, as well as all university coursers. The second hour provides group projects in the application of these productivity concepts to intradisciplinary case studies.

The case studies are written within an individual faculty member's area of expertise, and shared at weekly HAWK staff meetings. Thus the studies reflect the most up-to-date technology. Case studies involve a concept particular to the area to be discussed that week along with an exercise using said concepts. The productivity tools that have been mastered to that week are used in the solution to the exercises.

Students learn questioning techniques between peers and faculty in solving a variety of academic problems. Through written and oral reporting, they are strengthening their organizational and communicative skills.

HAWK faculty also serve as advisors to their students focusing the instructor as a learning and career development facilitator. Faculty and freshmen interacting on this multilevel basis suggest a basis for improved student retention.

ABSTRACT: A Writing Assignment for the Computer Literacy Course

David Olson, PhD, Western Oregon State College, Monmouth, OR

Computers and Society (CS 101) is required for all students as part of the Liberal Arts Core Curriculum at Western Oregon State College. Since an interdisciplinary approach to computer literacy was desired, a student writing assignment using the word processor, dealing with societal topics was included in the course requirements.

Although a writing assignment seemed like a good addition to the course, there were problems. Students have trouble transferring writing skills from a writing class to a computer class, and CS 101 instructors are not experienced teaching these skills. Some students object to losing points for poor grammar and spelling in a computer class.

In 1985 a series of assignments which resulted in a one-page essay on some societal topic relating to the course was developed with the help of writing teachers. Each of the four assignments included in the series can be presented and evaluated by instructors with limited experience in teaching writing. The four assignments include:

1. Full Thesis Statement
   Using a procedure described in *The Lively Art of Writing, Developing Structure* (Payne, 1982), students write a thesis statement. Using information found in their references, they write a listing of the pro arguments with citations and a listing of the con arguments to their thesis.

2. Rough Draft
   After the full thesis statement has been evaluated, students complete the rough draft of the paper. The rough draft must not exceed one page in length and must follow the format described by Payne.

3. Peer Review
   Students exchange two copies of their rough draft with other members of the class for peer review. In reviewing other papers, students are instructed to label both pro and con arguments and mark the paper for such things as redundancy, wordiness, and jargon.

4. Final Draft
   In addition the final draft, students also submit their two rough drafts, which have been peer-reviewed, and photocopies of the two papers they peer-reviewed for evaluation. Students receive points reflecting how well they reviewed other students' essays, how well they incorporated suggestions received on the two rough drafts in their essays and the quality of the final draft of their essay.

Since CS 101 is taught each term, each of the four assignments have been improved several times. The result is a format that results in several excellent essays each term.

ABSTRACT: A Design and Implementation of an Automated Test Generator

Marilyn P. Persson, 3804 Stetson, Lawrence, KS 66044, and Daniel A. Canas, Dept. of Computer Science, The University of Kansas, Lawrence, KS 66045

PMATH, a programmable mathematics test generator described in this research, is a system designed to aid math instructors in devising tests for math-related courses. This automatic test generator creates unique tests while allowing for user specification of the test content. The system provides an environment that allows for ease in adding new problems or entire courses to the existing system. A menu-driven user interface communicates user requests to the system.

Allowing for user specification of problem templates generating numerous unique problems is the main attribute of PMATH which distinguishes it from other test generators. Simplifying the task of inputting the constraints of a new problem template encourages the user to customize his/her tests.

For the unsophisticated user who wants to expend as little effort as possible, the system provides over 300 templates per text and predefined tests for each chapter. Generating tests from these system-defined test drivers requires that the user respond to only two questions. Much attention has been given to designing templates which avoid ambiguity, test important concepts, and generate a large selection of unique problems.

Flexibility and ease have been given to the user in specifying test content and format.

A high-level description of the design and implementation of the user interface will be presented. The method of specifying a problem template and test content will be discussed. System facilities which transform the user's specification of a test into a formatted test are described. Included in the research is a description of the mathematical text formatter provided by this system.

The system is currently being used at the University of Kansas to generate over 2,000 unique exams per week for testing students in college algebra. The course is taught using a self-instruction mastery approach where the average student retakes each exam several times before mastering it. In this teaching environment, it was necessary to have a test generating system to generate unique exams to prevent memorization of test content. Student and instructor feedback has confirmed that PMATH has been successful in generating sufficiently random tests which are a good test of the students' knowledge of the material.

Efforts to improve the quality and quantity of problem templates are ongoing. Environments where this generator could be used are recommended. Thoughts on the future development of this test generator will be presented.

ABSTRACT: Computer-Aided Instruction at The University of New Hampshire: Planning, Implementing, Evaluating

Colleen Kendall-Piel, University Computing,
Thompson Hall, University of New Hampshire,
Durham, NH 03824

As the result of a special allocation from the New Hampshire State Legislature, The University of New Hampshire's computer aided instruction (CAI) initiative, DISCovery, began in the fall of 1985. One year later, three microcomputer clusters containing approximately 100 AT&T 6300 and Macintosh Plus microcomputers opened and over 1000 students began

receiving CAI assignments in such disciplines as philosophy, zoology, sociology, and foreign languages. The second year of the project, saw the creation of a Resource center that includes a hands-on training classroom and a software library for faculty. Over the first 2 years DISCovery has funded 23 CAI projects involving 50 faculty from 26 departments.

This presentation will trace the first two years of the CAI initiative at UNH from the planning stages, through the selection, purchase, and installation of hardware, to the implementation and evaluation of faculty CAI development projects. Particular emphasis will be made on the specific steps taken to successfully implement the program, problems encountered, and the initial reactions of administrator, faculty, and students.

The discussion will focus on the belief of those involved in the DISCovery effort that the successful integration of microcomputers in higher education depends on more than just providing students with increased access to computers. It means little to say that one student in twenty at a given university has access to a computer without knowing anything about the quality of that access. The aim of the University of New Hampshire's CAI program, DISCovery, is to ensure a meaningful integration of computers into the educational process by coordinating the establishment of microcomputer clusters with the development of quality CAI. To do so requires, above all, the commitment and involvement of the faculty.

UNIVERSITY MICROCOMPUTER APPLICATIONS

John Fisher
Roberta Evans-Sabin
John Talburt, W. Steve Anderson
Nancy Engen-Wedin

ABSTRACT: Computer-Based Instruction in English Linguistics: A Demonstration and Discussion

John C. Fisher, Professor of English, State University of New York, College at Oswego, Oswego, NY

The controversy regarding the use of computer-assisted instruction has largely faded away, but not so the use of computer-based instruction, which can replace in at least limited ways the classroom lecture. Teachers' groups are rightfully concerned about the academic implications and the threat to the profession. A demonstration of a computer-based instructional program in English phonology, designed by the presenter, will focus attention on these concerns, as well as on the program itself.

The presenter will relate his limited experience with the program, which can be used to teach one-third of an English linguistics course to college students preparing to teach in elementary and secondary schools. Included will be a discussion of the likely advantages and disadvantages of computer-based instruction, and a comparison of classroom and computer lectures.

ABSTRACT: System Documentation Tailored to Student Needs

Roberta Evans Sabin, Assistant Professor, Computer Science Department, Loyola College, Baltimore, MD

Students entering a college computer science program most often have no experience with a time-sharing environment. Our entering students usually have completed a BASIC or general data processing course in high schools where micros are the only available computers. Like many colleges, we use a minicomputer (two networked VAX 785s) for instruction in introductory programming courses. Students use terminals located in several labs on campus, and increasing numbers of students use microcomputers to access the VAX from their residences.

A minicomputer as powerful as ours has many capabilities. Documentation supplied by the manufacturer, besides being expensive, is most often intimidating for beginning students. Students need to learn to use the system as quickly and, hopefully, as painlessly as possible.

To ease the transition to an environment where an editor and system commands must be mastered to run programs and print files, we distribute simplified documentation. An introduction to the system is combined with information necessary to enter and run programs. We use a series of booklets, each designed for a programmer of a specific language new to the system. Each contains, in nontechnical language, a very simple description of the hardware followed by an explanation of system commands needed to log onto and off of the system, use the editor, compile and run a program, access files in the system, and use the printer. Appendices provide more detailed information on system facilities, such as using the system debugger and submission of batch jobs. These booklets, each 20 to 30 pages in length and produced by instructors in our department, supplement the vendor-supplied documentation available to students in the main lab.

We have found that our in-house documentation decreases student anxiety and allows students to begin using the system immediately. We have found the investment of instructors' time in this project well worth the effort.

ABSTRACT: Preparing a Team-Taught Course in User-Documentation

John R. Talburt, Associate Professor, Department of Computer Science, University of Arkansas at Little Rock; and W. Steve Anderson, Department of English, University of Arkansas at Little Rock, Little Rock, AR

The presentation will focus on the pedagogical features of a team-taught writing course in user documentation: writing assignments, grading, textbook, and teaching methods.

We built the course around the program that we wanted documented, a locally written utility system (a panel generator) that had not been documented for lay users. All assignments were related to writing documentation for that program. Although we were responsible for grading all preparatory work, the final version of the user-documentation was evaluated by the potential users of the panel generator (in a test session, lay users programmed—or attempted to program—a

panel using documentation our students wrote).

At the time we were planning the course, texts for documentation seemed scarce, but at least two that were available seemed especially good. We finally chose a text that was strong on overview, but we supplemented liberally from a text that presented the modular system of writing software. The modular system, we found, allowed us to emphasize several important principles on writing documentation.

Further, students with computer science backgrounds could easily identify with the modular system, given its similarity to the top-down methodology used in software engineering.

In almost all cases, we found that the most effective method for teaching documentation involved "doing" rather than "telling." For that reason, we gradually moved through the term toward using more and more workshops. It was in the workshops that the presence of two instructors was especially helpful, although team-teaching was also important in developing the course concept.

ABSTRACT: Learning Disabled College Writers in a Microcomputer Enriched Composition Class: An Analysis of Performance Outcomes

Nancy Engen-Wedin, Research Associate, LD College Writers Project, The University of Minnesota, Minneapolis, Minnesota

Learning disabled (LD) students' writing can be jagged and difficult to read , there may be random errors in spelling and punctuation, and the syntax can be garbled. These writing problems are idiosyncratically compounded when held in conjunction with the emotional attributes of repeated failure; As a result, most LD students eventually drop out of writing classes and college programs all together. There is evidence that the use of microcomputer/word processing technology can be of some benefit for the LD student with writing problems. The purpose of this research is to study the impact of microcomputer or word processing technologies on the performance outcomes, especially the writing and course grades of learning disabled students maintreamed into required undergraduate writing classes.

Learning disabled and non-learning disabled students enrolled in a two course sequence of introductory writing classes, Personal Writing and Expository Writing, held at the University of Minnesota, General College. Classes met in a special writing lab that contains IBM, Apple IIe, and Macintosh microcomputers with word processing and spelling checker softwares available. All students generated writing samples at three times (pre, mid, and post) during the twenty week project. The writing samples were holistically, blind scored and evaluated for the total number of words produced and percent of misspelled words. Prior to and following the intervention all participants completed a writing apprehension inventory (Daly-Miller Writing Apprehension). Course grades and course completion rates were also obtained.

The project results are encouraging. Our first year findings were replicated and verified by the second year results. We found gains in most areas of writing outcomes for learning disabled students. Of most importance for our LD participants was a noted significant decrease in apprehension toward writing that occured as a result of the intervention. LD students also showed an increased rate of course completion compared to their NLD peers. Another significant finding that is especially pertinent to service providers is related to the pattern of gains that were made by the LD student participants. Though the intervention was 20 weeks in length, most of the performance and attitude gains that were significant appeared within the first ten weeks of intervention. All research results will be discussed in detail in the presentation.

The Learning Disabled College Writers Project is a three-year federally sponsored research/demonstration project funded under Postsecondary Educational Programs for Handicapped Persons - Demonstration Projects. It is supported by the University of Minnesota, General College and the Department of Academic Computing Systems and Services.

LOGO: ELEMENTARY

Dennis March
Paul Ohme, Martha Knight
Mary Upton

ABSTRACT: Making Movies with *LogoWriter*

Dennis March, Computer Coordinator, Clarion Community Schools, Clarion, IA 50525

LogoWriter is a program which combines the features of the Logo programming language with a word processing package. This program makes it easy to combine Logo graphics with text from a word processor. It was this ability that enabled students to make "movies" with *LogoWriter*. By movie, I mean that students drew pictures and wrote text to go with each picture in order to create separate pages. A short program was written that would automatically flip the pages. While the computer was flipping the pages, a VCR was hooked up to the computers to record. You now have *LogoWriter* movies saved on tape.

Some of the project objectives are:

1. Learn some new *LogoWriter* commands;
2. Work on idea development and planning skills;
3. Learn about how a computer and VCR can work together; and
4. Promote individual creativity.

The students were eighth graders in an exploratory computer class. All eighth graders had this class every day for one quarter of the school year. Each student had at least four weeks of work with Logo the previous school year. Following the review of previously learned commands and an introduction to some new features of *LogoWriter*, students were given a planning sheet on which they designed each page of their short movie. The movie was required to consist of at least six pages. After this planning stage, the students' plans were checked by the teacher to see if they could realistically be done within the assigned time frame and for grammatical errors in the text. The project took approximately six days to complete.

The project has served as an exciting way for students to review the things learned previously, as well as learn some new features of the *LogoWriter* program. The tape of the movies served as a fine public relations device as it was shown at the school's open house.

ABSTRACT: Improvement of Critical Thinking Skills with Logo at the Second Grade Level

Paul Ohme, Northeast Louisiana University, Monroe, Louisiana and Martha Knight, Concordia Parish Schools, Ferriday, LA

Computers and the technology which produced them have had and will continue to have a profound effect on society. The power and versatility of microcomputers make them a learning tool, not only for drill and practice but also for critical thinking and problem-solving.

Professional educators have responded to this emergent technology and have accepted it, not as a passing trend, but as a permanent place in education. Therefore, students at an early age should be introduced to microcomputers.

In June, 1987 Ferriday Lower Elementary School received a grant from the Education and Community Resources Program from the International Paper Company Foundation. The goals and objectives of the funded project focus on critical thinking skills with the use of *LogoWriter* to enhance both mathematics and English/language arts at the second grade level. Eight second grade teachers underwent an intensive inservice training program coordinated between outside consultants provided by the grant and Northeast Louisiana University personnel funded by Title II. After noncomputer preparatory activities, second grade classes were scheduled in the computer lab for two fifty minute periods per week. Limited equipment and no supplemental teacher support gave rise to a variety of teaching techniques. Through the cooperation of the local public library and grant funding students are able to accompany their parents and other relatives to the library during nonschool hours to work with their *LogoWriter* work disks. This activity is encouraged by the teachers assigning homework to be done at the library. Parents from this predominantly minority population are making major efforts to get their children to the library.

The success of this project is creating a morale boost for the entire faculty and encouraging teachers of all grade levels to participate in computer training. Empirical observations indicate that the project is achieving all of its intended goals. The project coordinator hoped to able to do formal research next year to verify this improvement in critical thinking skills. To date very little has been published concerning the use of Logo in the primary grades.

ABSTRACT: University of Virginia's *Surveys* and *GraphTools* programs to be used with *LogoWriter*

Mary Upton, Assistant Principal Jackson Elementary,
Lubbock Independent School District, Lubbock, TX

With *LogoWriter* and its unique features of graphic and text manipulation, students have easy access to the "problem-solving microworld" so often associated with Logo. With the addition of *Surveys* and *GraphTools*, students enter the "microworld" of graphs. Students can make and take surveys via computer. Then the results of these or other graphs can be illustrated in the form of a bar, line or circle graphs.

Surveys and *GraphTools* consists of four separate disks containing tool programs to be used with elementary and junior high students. The following is a brief discription of each disk:

1. Students can design, take and print surveys with the Surveys disk. An overview of the entire project is included on this disk and can be printed at the users convenience.
2. Students can design simple bar graphs or comparison bar graphs. The comparison bar graphs use a colored bar compared to a white bar.
3. Students can design simple line graphs or comparison line graphs. The comparison line graphs use a colored line compared to a white line.
4. Students can design pie graphs. By simply entering the number of items for each category, a pie graph will be drawn.

The three *GraphTools* disks each contain an instruction page that can be printed. These instructions will lead the user through the three steps in using each disk. Students can do a tutorial first to see how to make each graph and what each number means. Next, a tool program is available for students to enter numbers as prompts are given. After the student has worked with this for awhile, he/she will be ready to use the tool procedure that require numbers as inputs to the procedure. These tool programs can be loaded into memory and used to draw graphs without being saved on the flip side of the page.

Surveys and *GraphTools* are versatile, easy-to-use programs created during the 1987 Logo Fellowship at the University of Virginia. They can be used only with *LogoWriter* and on Apple machines. Information about obtaining copies of these public domain disks will be availiable.

RESEARCH IN THE APPLICATION OF COMPUTERS IN EDUCATION: TRENDS AND ISSUES

Henry Jay Becker
Johns Hopkins University
Baltimore, MD

Betty Collis
Dept. of Psychological Foundations
University of Victoria, Canada

Jef Moonen
COI, Twente University
Postbus 217
7500 AE Enschede, The Netherlands

Decker Walker
School of Education
Stanford University
Sanford, CA

ABSTRACT

A tremendous volume of research in the application of computers and computer-related technology in education is occurring throughout the world. In this presentation we employ two perspectives - geographical and methodological - to identify some current trends in this research and to comment on issues relating to the applicability of this research to practice and decision making in the educational setting.

Approaches to a Synthesis of Research

There are many different perspectives that could be used to organize research on the application of computers and computer-related technology in education. A subject-matter perspective could be used, which would expedite focus on the research for individuals particularly interested in curriculum and instruction in particular areas such as mathematics or social studies. An age-group categorization could be more appropriate for those who deal with a cohort of students in a variety of different classroom focuses, as is the case with the preschool or elementary school practitioner. Other organizing frameworks could include a focus of type of computer application; or on type of outcome; or on contextual variables; or on some distinction between "basic," cognitively oriented research and field-based evaluation research. Another perspective of interest is a geographical one, both for descriptive and comparative purposes.

Research From a Geographic Perspective: Western Europe

In Western Europe, most countries are introducing computers in education following national policies relating to this field. In some countries political changes have a major impact on the policy. Most countries are conducting evaluative research of their policy. Besides government influence on research questions, academic priorities are of major importance in the choice of research areas. These priorities often relate to developmental work using the newest technologies or knowledge-based strategies, comparative research exploring variation in instructional strategies, and process-oriented research to strengthen insight in the theoretical aspects of teaching and learning. Finally, international organizations with a strong European basis, such as the Council of Europe, UNESCO, OCDE, and the Commission of the European Community, are very interested in the impact of new information technology in schools and contribute, each in their own way, to the stimulation of research programs. From these perspectives, major trends and issues in Western Europe will be described as the first component of our presentation.

Research from a Methodological Perspective

Another and somewhat unusual approach to a research synthesis could be one which uses methodological strategies as a basic framework for organizing a commentary on research. A benefit of this type of approach is a focus on strengths and limitations of different research approaches and the impact these strengths and limitations can have on applicability of the research to the classroom teacher or system-based decision maker. This is the approach we will use for the remainder of our presentation. The following points will be discussed.

What Surveys Can and Cannot Tell Us

Surveys have a particular value in overcoming our natural inclination to focus on unrepresentative settings in which some schools parlay resource advantages (sophisticated teaching staff, bright student body, high ratio of computers-to-students) into splashy anecdotal news, as well as the inherent disadvantages of using these unrepresentative settings as a basis for actually measuring the consequences of computer use. Changes in focus in major survey studies involving the use of computers in education since 1983 will be discussed.

What Do We Learn From Experiments?

"Typical" experimental studies will be described and problems associated with them - problems related to smallness of scale, shortness of duration, or compromised design - will be highlighted. In contrast to these, still-economical large-sample experiments that study implementations of computer-use practices in typical settings over a full school year will be described. This contrast will be used to illustrate why we should take a very critical approach to claims made on the basis of existing studies, and will provide a framework for a model for conducting field studies of typical practices.

What Generalizations Can We Make From Quasiexperimental Research?

Quasiexperimental research includes research situations where we must assign intact groups to treatments rather than randomly assign individuals to treatment. The confounding variables that can influence the results or generalizability of this type of study are well known, yet this approach is frequently the "best we can do," given limited control over real-world settings and subjects. Examples of strengths and limitations of some current research studies of this type that relate to computers in education will be discussed.

What Can and Can't We Say, Based on Correlational Data?

The problems of making causal conclusions about correlational data are also well known, yet many practitioners may ask what real use can be made of associational studies if we do not move beyond a description of relationships. Examples of structural modeling as a way to test causal hypotheses about correlational data will be highlighted in terms of their practical value to the user of research in the field.

What Are Trends in Studies Using Qualitative Data?

A goodly number of qualitative studies of the use of computers in education have been reported in the past decade. This portion of the presentation will identify the problems addressed in qualitative studies as well as the conclusions reached. The review will provide an occasion for examining the appropriateness of qualitative methods to the study of computers in education and the strengths and limitations of these methods in comparison with other methods for addressing the major current problems in the application of computers to education.

Conclusion

Does categorization by methodology provide a useful way for practitioners to look at research on the application of computers and computer-related technology to education? The presentation will conclude with a consideration of this question by all the panelists.

TELECOMMUNICATIONS

Kenneth Epstein
Cynthia A. Myers, Vivien Carver
Karl Rodabaugh
Charlotte Scherer

ABSTRACT: Making Local Area Networks Useful Instructional Networks

Kenneth Epstein, Assistant Dean, Gallaudet University, School of Preparatory Studies

Local Area Networks (LANSs) present intriguing possibilities for instructional computing. However, as with most applications of computer technology, actually putting a LAN to work in support of instructional programs is not as simple as it sounds. The Gallaudet University School of Preparatory Studies has been experimenting with instructional networks using off-the-shelf hardware and software for three years. During that time we have had to address a wide variety of technical and practical problems. We have some solutions and a number of remaining questions.

The IBM based LAN consists of 10 student stations and one instructor station. The student stations are IBM PC/XT double floppy disk systems with 640 Ks of RAM each. The instructor station is an IBM XT with one floppy disk, one 20 Mb fixed disk, and 640 Ks of RAM, upgraded with an 80286 accelerator card and disk cache software. The LAN hardware and software are 10-NET by Fox Research. Additional video display control is provided with CT-System III by CompuTeach of Washington, DC.

The network is used by math, English, and business software classes. The math programs include basic arithmetic skills, basic college math, algebra, and geometry. English classes use the network to practice writing skills with real-time mail/phone software. Introduction to computers, word processing, and database management classes are also taught on the network. The network is used for both teacher-directed instruction and by the students on their own for homework and self-directed practice.

The Macintosh network is actually a printer-sharing system rather than a true local area network. It consists of eight Macintosh SEs with two floppy disk drives each, four Macintosh SEs with one floppy disk drive and one 20 Mb fixed drive each, one Macintosh II with one floppy disk drive and one 40 Mb fixed disk drive, one Imagewriter printer, and one Laserwriter printer. The equipment is linked together with AppleTalk. The Macintosh network supports word processing and spreadsheet software as well as an experimental geometry tutor.

In order to make the network technology useful for instructional applications it has been necessary to create a system with several critical characteristics. The system must be easy for faculty and students to use. It must not be intimidating. The system must be reliable and easy to maintain. The system must be flexible and adaptable to a variety of teaching needs and styles. The system must be accessible and available. This presentation will describe the hardware, software, and administrative steps that we have taken to create useful and well-used instructional networks.

ABSTRACT: Free-Net: An Open Access Information and Education Network

Cynthia A. Myers, M.A., Director, Health Education Center, St. Elizabeth Hospital Medical Center, Youngstown, OH; and Vivien Carver, Ed.D., Associate Professor, Department of Health and Physical Education, Youngstown State University, Youngstown, OH

The Youngstown Free-Net is a free, open-access community computer system available to anyone with a microcomputer and a modem. A computer located on the campus of Youngstown State University has been programmed to serve as a community information and communications resource, providing access to a wide variety of electronic services and features.

Free to users in much the same way that a public library is free, Free-Net is a community resource that allows users to dial into the system and post questions, review information, and receive answers. Some of the services and features available on Free-Net include online medical and dental information, online communication with the Mahoning Valley Poison Center, and electronic contact with government officials.

Youngstown Free-Net is only the second free, open-access, community-operated computer system in the nation, and is modeled after the Cleveland (Ohio) Free-Net. Co-sponsored and operated by Youngstown State University and St. Elizabeth Hospital Medical Center (a 750-bed community teaching hospital), Free-Net currently has approximately 2,000 users online.

Plans are currently underway to link the Youngstown City School System to Free-Net, a project that will bring some 80,000 local students online. One important feature of this endeavor will be "Ask-A-Doc," in which a resident physician from St. Elizabeth Hospital Medical Center will be available for a two-hour period once a week to answer specific questions online from students.

Direct all correspondence to Cynthia A. Myers, M.A., Director, Health Education Center, St. Elizabeth Hospital Medical Center, Youngstown, OH 44501.

ABSTRACT: Computer Technology and the Delivery of Non-Traditional Education: Current and Potential Applications of the Optel System

Karl Rodabaugh, East Carolina University, Office of

Conferences & Special Programs, Greenville, NC

Nontraditional students who wish to take college courses or enlist in professional development programs frequently have a limited number of options. One, they can drive to a nearby college or university. This is a good option if a college or university is nearby. Two, they can rely on a college or university to bring the classes or programs to them. This doesn't always happen. Three, they can take some sort of correspondence course, purchase a "canned" training program, or participate in a televised professional development program. But these courses and programs are not always suitable. In numerous instances credit will not be received when transferring such courses into a degree program and professional groups often will not certify such "unofficial" training programs.

A fourth option is available. Advances in computer technology have created numerous possibilities for the delivery of college-level classes and professional development programs directly to any school or business in the country, no matter how isolated. Between the satellites and the telephones are other systems which rely on computer technology to offer both interactive voice and graphics, and which are proving to be excellent mediums for teaching and training. In addition, these systems offer solutions to the problems of costs and distances.

A special example of the intermediate computer-driven delivery modes is the Optel System, which consists of a writing tablet toegether with an IBM-compatible computer linked to other units over a single telephone line. Real-time voice and graphics interaction are possible, with still-frame video transmission available as an option.

One exciting example of the system's potential is evident in the link between the Boston University School of Medicine and China's Beijing University. Medical courses are being taught to the Chinese over this system. If Boston University can apply computer technology to teach medicine to people in China, what is to prevent any university or any corporation in this country from delivering credit courses or professional development programs to nontraditional students at any location in the United States or the world?

ABSTRACT: Electronic Pen Pals: Education Majors Converse with Junior High Students

Dr. Charlotte Scherer, Director, Clinical & Computer Labs, College of Education & Allied Professions, Bowling Green State University, Bowling, OH

Secondary Education majors are learning about telecommunications and gaining insights about junior high students through an electronic pen pals project with the local junior high students. The Computer Resource Center at BGSU, which serves schools in the Northwest Region of Ohio, has set up a free electronic bulletin board, called the "Buzz Board", for use by their member districts. Volunteer secondary education majors have been matched with students in the Computer Literacy classes in the local junior high school and are writing back and forth on-line using the electronic mail portion of the Buzz Board.

The project was piloted in the spring with just three secondary education majors communicating with whole classes at the junior high. In the fall, the scale was enlarged to 15 secondary education majors, and 70 junior high students, with each college student writing to approximately five or six junior highers. Both sets of students have to learn something about telecommunications in order to access the system and participate in the project. Plans call for the students to eventually meet their electronic pen pals face to face near the end of the semester.

Goals for the project are as follows: (1) to provide education majors with (a) insights about what junior high students are like, (b) experience in using computers for telecommunications, and (c) opportunities to see "real world uses" for computers; (2) to provide junior high students with (a) an opportunity to use telecommunications; (b) an adult person with whom they can share their ideas, interests and problems; and (c) information about a variety of ways computers are used in the "real world"; and (3) provide possible recruitment of junior high and high school students for secondary education microteaching programs.

Results of the project during the fall semester indicated that the education majors did gain insights into junior high age students. Some were surprised at the ease with which eighth graders used computers, that they would "open up" so readily to college students, and at how involved socially junior high's are even though so young. Education majors also felt they gained some competence with computers and modems, but would have liked more experience. The junior high students enjoyed contact with college students, were interested in college life and were unsure at first what to say on-line to the students. They enjoyed meeting their pen pals face to face. The project director found that training was time consuming, as was management of messages. The project will be refined and continued another semester.

BUSINESS APPLICATIONS

Helen Ligon
Gary McDonald, Merry McDonald
Andrew Schultz, Mary Sutton, Barbara Zigalo

ABSTRACT: Using Fourth-Generation Query Languages in the MIS Class

Helen H. Ligon, Professor of Information Systems, Baylor University, Waco, TX

Managing information is as important as managing the other business assets--men, machines, money, and materials. An efficient method of managing information is through the use of fourth-generation query languages. FOCUS and ORACLE are the two languages selected, with FOCUS used on the personal computer and ORACLE used on the mainframe.

Because many students taking the MIS class are *not* Information majors, they have not had exposure to querying databases or files. Such an ability is becoming more and more important to today's managers and future managers. In addition to giving the student the concept of ad hoc queries and short reports, using these languages give students experience in both using the personal computer and the mainframe for both decision support and executive information support.

For the FOCUS assignment on the personal computer, students are given handouts containing elementary syntax for the query language. One of the projects asks them to actually create a small database and then query the data in several different ways. These queries quickly create reports and give students the appreciation of speed and depth of FOCUS for ad hoc reporting.

To expedite their learning and curtail the time-consuming portion of database creation, students are also given a completed database and asked to query the information it contains.

As the assignment is presented, FOCUS syntax is discussed in class with the student being exposed to the way in which FOCUS generates the database. Also included are specific instructions for accessing the FOCUS package and the various parts of the software.

Many business students have very little mainframe exposure other than through their use of statistical packages in a very structured environment. The departmental faculty felt that students in the MIS course should have some mainframe exposure since many of them will work with terminals in a mainframe environment when they enter business.

ORACLE is a very extensive database language, but the MIS students are not expected to learn all of the nuances of the language. Instead, they are assisted in learning how to obtain selective reports from a database that has already been created. As with the FOCUS asignment, the student is given a handout, classroom instruction in the syntax ORACLE, and a set of queries to complete.

Student response has been excellent. They appreciate learning a fourth-generation language when they can use it first-hand. In addition, many prospective employers have told us that they are widely using both FOCUS and ORACLE to assist their employees in querying the company database for needed information.

ABSTRACT: A Database Course--Integrating Theory and Practice

Dr. Gary McDonald and Dr. Merry McDonald, Professors of Computer Science and Information Systems, Northwest Missouri State University, Maryville, MO

Northwest offers a one-semester upper-division database course covering the design and implementation of databases, with emphasis on the relational model. The course is required for Computer Management Systems majors and is an elective for Computer Science majors. We have two main objectives in the course:

1. To present the fundamental concepts of database theory; and
2. To give students practical experience in the design and implementation of a database.

Objective 1 is achieved through lectures and assigned exercises. To achieve Objective 2 students organized in teams are assigned a project which is completed in stages, starting with client interviews to determine requirements and ending with an implemented database using R:base 5000. Work on the project is done outside of class with the students consulting tutorials and manuals covering the relevant features of R:base

5000. We feel this is a significant learning experience for the students that simulates the work environment they will soon encounter.

Experience has shown that there are two difficulties with this approach:

1. Students' lack of experience: For most students, this is the first attempt to design a system in an open-ended, unstructured environment. Errors in early stages propagate through the design process and result in poorly constructed implementations; and
2. Limited knowledge of the software package being used drives the design process: If students do not know how to implement a feature, they find it easier to omit it from the design than to consult manuals and experiment with the software to obtain a solution.

We will present projects, including handouts, which we currently use in the course and discuss our approach to alleviating these problems.

ABSTRACT: Integrated Software: A Comparison of Three Popular Packages

Barbara Zigalo, Microcomputer Consultant, Silicon Systems, Inc., Tustin, CA; Mary Sutton, Microcomputer Consultant, Computer Center, California State University, San Bernardino; Andy Schultz, Asst. Professor, School of Education, California State University, San Bernardino; and David Neighbours, Associate Director, Computer Center, California State University, San Bernardino, San Bernadino, CA

How can a microcomputer help me control student information, grades, correspondence and other requirements for class management? Will an Apple II, a Macintosh or an IBM PC best suit my needs? Where can I begin in answering these questions?

Right here! This presentation will help you save sleep and sanity in finding answers to the questions above. During this presentation you will see three popular integrated software packages on three popular microcomputers. *AppleWorks* for the ubiquitous Apple II, Microsoft *Works* for the Macintosh, and Microsoft *Works* for the IBM PC. Each integrated software program will be able to solve many of the class management problems at the same level of complexity, but you may find a difference in speed, ease of use, and flexibility.

How integrated software can assist educaters with the preparation of professional documents, class gradebooks, correspondence, and student database will be discussed. In addition, the advantages and disadvantages of integrated software and computer systems will be examined.

GETTING STARTED IN ARTIFICIAL INTELLIGENCE: A LIBERAL ARTS PERSPECTIVE

R. Waldo Roth
Taylor University
Upland, IN

ABSTRACT

Although artificial intelligence (AI) has existed as a research discipline and academic field for some time, the advent of powerful, inexpensive workstations and PCs has brought AI applications into the realm of possibility for small colleges. The problem is that with already exploding curricula, some liberal arts colleges are struggling with what, if anything, they can add to their programs. Nevertheless, the interdisciplinary aspects of AI make it a very appropriate and attractive discipline to offer in a liberal arts college, given a qualified faculty and some moderately priced equipment.

This session will deal with how one liberal arts college started such a program six years ago, mush of which is transportable to other liberal arts programs. Topics such as curriculum development, course content, equipment options, and steps to pursue funding will be addressed.

SPONSOR: ASCUE

NEW METAPHORS FOR PRODUCING TECHNOLOGY BASED LEARNING MATERIALS

Alfred Bork
Educational Technology Center
Information and Computer Science
University of California
Irvine, CA

ABSTRACT

So far the use of the computer in education has had only a minor effect on the educational systems of the worlds. We continue to have serious problems everywhere in education. If anything, the computer has increased these problems, rather than helping to solve these problems. The courses we now teach, even in locations with many computers, are essentially the same courses we taught in the precomputer age. These courses are often antique in content, and they seldom work for all students. Students are typically passive.

Effective use of computers in education at all levels demands that we produce high quality highly interactive learning material using the new media, computers, videodiscs, and compact disks. So far much of the material produced has been of poor quality, and the amount of material available in any one area has been too small to have any sizable impact on our courses. We need more full courses that assume the technology is available as the course is developed.

At the heart of this difficulty is the use of ineffective processes for producing technology-based learning modules. Simplistic authoring systems, leading to simplistic material, have been widely promoted; vast sums of money have gone into developing these systems, but little is produced. Or the organizational structure is not up to the task. Producing curriculum material in any medium is a demanding task if it is to be effective, suggesting that a well thought out organizational structure is needed. Efforts in schools and universities are often naive in considering the amount of work and money that will be needed for success. The scale of development is typically too small to have any impact on education.

Hence we need to reconsider how we are to produce the technology based learning materials we require if we are to make substantial improvements in our educational system. Some assistance in this task if offered by examining major centers of curriculum development around the world.

Perhaps the outstnading center is the Open University, in the United Kingdom, located about 50 miles north of London. So far technology has played only a minor role in the extensive curriculum efforts here, but the Open University has a firm grasp of the strategies needed to produce effective learning material, regardless of the media involved. Other similar centers also give us clues about how to proceed.

But producing computer-based learning material is not identical to producing material for books, video, and other noncomputer media. The key to the new media is interaction. We need new strategies, new metaphors, for developing material unique to the computer. This talk explores metaphors and models that look promising. These ideas are not intended to be exclusive; many might be involved in a single project.

The ideas to be discussed are the office model, the artificial intelligence model, the hypertext model, the market model, and the systems approach. Each model or metaphor can have special tools that aid the production process. In each case prospects for future developments are also commented on. These strategies begin to suggest how we could make major changes in education with extensive use of the technology, changes that would lead to an improved educational system.

These production approaches need to be considered in the context of full-course development. They will lead to new types of courses, structures not possible with our older learning technology. The notion of hypercourse will be introduced in the talk as an example of such a new course structure.

A COMPUTER-AIDED ADA COURSE

M. M. Tanik, B. Shirazi, J. Sampson
Department of Computer Science and Engineering
Southern Methodist University Dallas, Texas

U.W. Pooch
Computer Science Department
Texas A&M University
College Station, Texas

ABSTRACT

Reconfigurable, project-oriented, computer-aided Ada courseware [1] that has been developed and successfully used in an academic environment is described in this paper. The courseware can be used for an undergraduate ACM-CS1 level programming course or as an introductory industrial Ada training package. It is designed to teach not just the Ada programming language, but also problem solving techniques and programming methodologies. These subjects are addressed individually to facilitate the student's grasp of each topic. An attempt has been made to match the proposed courseware's objectives with those stated by the ACM Curriculum Committee Task Force in 1984 [2]. In addition, ongoing efforts in augmenting the courseware with an intelligent help system, similar to the one developed for the IBM CMS operating system, are summarized [3].

INTRODUCTION

Early research on developing self-paced courses identified features such as: 1. individually paced; 2. mastery-oriented; 3. use of study guides for communication of information; and 4. lectures for the purpose of stimulation and motivation [4,5]. In developing this computer-aided Ada course we included most of these features. In addition, we used ACM Curriculum recommendations as a course development guideline.

The objectives stated in the ACM recommendation can be summarized as follows:

1. To introduce a disciplined approach to problem solving methods and algorithm development: One of the most fundamental CS1 objectives is that of problem solving. ACM suggests teaching such problem solving methods as the problem decomposition strategy, solution by analogy, solving a special case of a general problem, and conversely, generalizing a specific solution. A student must be taught to analyze a problem by first understanding it, and then specifying input data and output requirements.

2. To introduce procedural and data abstraction: Along with problem solving, algorithm development needs to be thoroughly discussed. This should include the use of pseudocode and pictorial descriptions in representing algorithms, and desk checking and test data used in the verification of algorithms. Students should be taught the importance of procedural and data abstraction. Extensive use of subprograms must be encouraged in student programming assignments. Various concepts relating to subprograms, such as procedures and functions, parameters and identifiers, subprogram nesting and recursion must be introduced. In addition, sequential, iterative, and conditional control structures must be taught. The introduction of various primitive data types and their relational operators, along with structured data types, will help the student to employ data abstraction in their programs.

3. To teach program design, coding, debugging and testing, and documentation using good programming style: The ACM divides program development into four areas, the first being that of design. A good program will start with top-down design and stepwise refinement, and employ procedural and data abstraction, and information hiding. The next area is coding, which includes the use of structured control statements and good programming style. Program correctness is another element in development. The student must learn to generate test data, along with the use of output statements, stubs, and drivers to help in debugging. Lastly, documentation techniques must be addressed.

4. To teach a block-structured high-level programming language: The programming language taught in the CS1 course is an important concern. It must be one that can allow a student to use what is learned during the CS1 course. It should be a block-structured high-level language with such features as structured control statements, structured data types, strong data typing, subprograms with parameters, dynamic storage allocation, and recursion. Typical languages include Pascal, PL/I, and Ada.

5. To provide a familiarity with the evolution of computer hardware and software technology: Another important subject area is that of the development of computer hardware and software technology. This includes the history of programming languages, from machine to assembly to high-level languages, and the translation of these high-level languages. The internal representation of data types is also covered.

6. To provide a foundation for further studies in computer science: The ACM also lists advanced topics that may be introduced in the CS1 course. Their coverage depends on both the students and the teacher. These topics include advanced data structures, searching and sorting algorithms, the analysis of algorithms, and program verification with loop invariants.

This Ada courseware is designed with the above objectives in mind. Programming is introduced as a means of solving the problems and implementing the algorithms with great attention paid to good program design, coding, debugging, and testing, as well as programming style and documentation. As the course name indicates, Ada is used as the programming language. This provides us with a suitable tool to teach procedural and block-structured programming, and introduce data abstractions. In addition, the course covers an overview of computer hardware and software technologies. Computer hardware components and their functionality are studied; operating system concepts are introduced; and other system software such as editors and compilers are presented. CS1 students are expected to have been exposed to computers and programming before. A strong mathematics background is highly desirable [6]. In order to support analysis of algorithms presented in CS1, the students are recommended to take a discrete mathematics course.

Our four credit hour CS1 course includes two hours of laboratory time. The laboratory time is essential to the smooth running of the CS1 class. Students learn the proper use of available computer resources. Quizzes are given, along with other material to complement the lecture class. Students are helped with their programming assignments to the extent that programming tools such as editors and debuggers are taught. Additional time is usually spent by the student to complete programming assignments.

The Ada-centered courseware includes a complete set of programming assignments, daily labs, and instructor directions [1]. Assignments start out short and relatively simple, and become longer and more complex as new concepts are introduced, and the student becomes comfortable with the programming environment.

The computer facilities available must support the programming assignments and labs. These resources should be in an interactive programming environment, such as that found with a time-shared mainframe or minicomputer, or a personal computer. This courseware has been developed for use on a microcomputer, such as the IBM PC using Janus Ada, or it can be used on larger systems, such as a Vax 11/780 with Verdix Ada 5.41.

COURSE ADMINISTRATION HIGHLIGHTS

A student taking this course discovers that the coursework is divided into 42 units. The content of the units corresponds to conceptual chunks in programming languages and Ada, and is consistent with ACM Curriculum 1984. At the start of the course, students receive a study guide and course syllabus to direct the work on the first unit. When we administered the course, we either used printed guides or directed the students to computer terminals that included necessary documents in their accounts. Before progressing into the next unit in the course sequence, students demonstrate their mastery of the current unit in the laboratory by performing the necessary work.

Faculty experimented in teaching Ada using this courseware in academic environments and our course formats reflect this experience. However, the approach we have taken is adaptable for teaching professionals as well. In teaching a course such as this for an industrial audience, the emphasis should be put on the lab activities and self-paced nature of the course. The industry professionals are expected to have the required maturity and motivation to successfully complete the course on their own with minimal guidance from the instructor. The lab activities are more useful for these professionals in order to emphasize the practical side of this course.

COURSE SUBJECT AREA HIGHLIGHTS

The first three lessons are introductory in nature. They include such topics as the objectives of CS1, the operating system, the editor, the development and function of computer hardware, the history of Ada, and the difference between compiling, linking and running a program. The Daily Labs that accompany these lessons allow the student to become familiar with a computer and the use of an editor. In addition, students are taught how to compile, link and run a sample Ada program.

The next set of lessons, up to the first test, introduce many important concepts. These include: top-down design with stepwise refinement; reserved words; identifiers; commenting; subprograms; the writing and use of an Ada Package; variables; data types; parameters; and the first control structures. The daily labs accompanying these lessons include programming assignments that use the introduced ideas including the creation of an Ada package to be re-used throughout the duration of the course.

The following lessons address problem solving and the use of control structures in depth. They also look at documentation, structured programming, top-down design, pseudo-code, algorithms, functions, and recursion.

During these Lessons, the students begin working on the first major project. The appendix presents an example of a program covered in Lesson #13 which concentrates on control structures.

Next we focus on constants, ranges, types, one and two dimensional arrays, searching and sorting, debugging, and the use of external data files. Students begin the second project during this set of lessons.

The remaining portion of the course focuses on the completion of all exercises and any literary items that have not already been covered. A list of literary items at the end provide guidelines on topics to be included. These are the history of computers, major hardware components, system software, evolution of programming languages, translation of high-level languages, internal data representation, major languages and their uses, security and privacy, data communications and networking, and microcomputers.

COURSE FORMAT

Although it is most important to cover the right subjects in the courseware, the format of the course plays an important role in successfully meeting the objectives. We feel that the format of this courseware is one that will allow students to fully grasp the concepts taught and it is consistent with our objectives stated earlier.

The courseware starts out with a syllabus. This lists the lesson number, topics to be discussed during the lectures, and the labs to be used with the laboratory section of the class. The course is designed so that one lecture period and one laboratory period comprise one lesson. There are 42 lessons in the course.

Lesson plans are distributed among the students in order to provide a guideline of the material to be covered. The lesson plans include daily lab information to be completed with each lesson and contain a set of useful appendices. In the courseware manual [1], there are instructions for copying the daily lab, and the manner of using it to prepare for the next class or lab. In addition a problem solving methodology is described in the courseware manual. It leads the student step-by-step starting with the student stating the problem in his own words and dividing the problem into smaller sub-problems, al the way to the coding.

The daily labs are designed to reinforce concepts that have been taught in class, while teaching the student problem solving and programming. The labs start out instructing the student in the use of the editor and operating system. However, they become progressively more difficult as the student becomes better acquainted with programming and the programming environment. This is accomplished by giving less and less of the problem solution, until the latter section of the labs when the student is given only the problem statement.

This approach allows the students to become confident with the programming environment by allowing them to be successful in their attempts from the beginning.

The daily labs are administered in a four-step process. First, the student copies the lab and the approved solution from a master diskette or file directory before the class meets. During class, the instructor presents the lesson, and the student is allowed to ask questions about the lecture. The students then run a copy of the approved solution of the lab to see the correct output format and observe that the daily lab can be done. The students proceed from that point on with the lab during the remainder of the laboratory period on their own.

Programming projects are assigned after the daily labs have been completed. The projects are designed to make students synthesize the concepts they have learned in a block of lessons.

In addition to daily labs, the student is given an objective, such as lesson 13 in the appendix, and a complete problem statement for each project exercise. The problem statement includes a complete background, requirements, and extensions to make the project more challenging. Each student also has access to an executable object file for the assignment that the student can run to see what the actual output should look like.

FUTURE IMPROVEMENTS AND THE USE OF AN INTELLIGENT HELP SYSTEM

We feel that an incremental project-oriented courseware designed around Ada is one of the best ways to meet requirements for a CS1 course described by the ACM Curriculum Committee Task Force of 1984. Ada not only provides the language constructs needed to teach this course, but also emphasizes the need for code reusability. Ada also allows tasking, real-time programming, simulation, and scientific calculations, making it usable throughout a student's computer education. In addition, the Ada language is completely standardized. Students can learn Ada, and know that they will not have to learn different versions of it when it is implemented on different machines.

The success of this incremental Ada-based programming course will increase at least an order of magnitude if an intelligent help system is provided in the background to support students' needs for domain-specific knowledge. This automatic help system will provide us the means to implement Keller's teaching plan [4], (individually paced, mastery-oriented approach) in our courseware.

There is a widening gap between the novice students and the sophisticated operating systems and Ada support environments. Because of this gap, the students, who are simply attempting to run a specific program, get frustrated and discouraged. The problem of interfacing with novice users usually demands more intelligence and friendliness from the system. Research is in progress [7] to provide intelligent assistance for novice and expert users of this courseware. This intelligent help system is to be similar in concept to the one developed by D. Yun [3] for the IBM CMS operating system.

CONCLUSION

Over 200 pages of course material has been developed and class-tested for three years at Texas A&M, the University of Texas at Dallas, and Southern Methodist University. It has been our experience that this package helps to take some of the load off the instructor, allows students to go at their own pace, and enables students to get on-line help on their own. As universities and industry switch to Ada, this kind of a computer-aided approach in teaching the Ada language and concepts can help make the transition from other languages much smoother.

ACKNOWLEDGMENTS

Some of the early research in developing this courseware was performed by U. Pooch and S. Teel at Texas A&M University. Later improvements were done by U. Pooch and M.M. Tanik at University of Texas at Dallas. Augmenting the courseware with an intelligent help system was suggested by D.Y.Y. Yun. Work at Southern Methodist was done with the assistance of K. Burns and J. Abuzaid. In addition, we would like to acknowledge the improvements suggested for presentation style and clarity by anonymous referees.

REFERENCES

[1] M. M. Tanik, U. Pooch, Ada Courseware Manual, Unpublished work, 1986.

[2] E.B. Koffman, P.L. Miller, and C.E. Wardle, "Recommended Curriculum for CS1, 1984," *Communications of the ACM*, Vol 27, Oct. 1984.

[3] D.Y.Y. Yun, D. Loeb, "The CMS - HELP Expert System," *Proceedings of the International Conference on Data Engineering*, 1984.

[4] J.A. Kulik, C. Kulik, and K. Carmichael, "The Keller Plan in Science Teaching," *Science*, Feb. 1974, pp. 379-

383.

[5] R.M. White, D. Pessel, "A Self-Paced Course in Introductory Computer Programming," *IEEE Trans. on Education*, Nov 1975.

[6] A. Ralston, "The First Course in Computer Science Needs a Mathematics Co-requisite," *Communications of the ACM*, Vol 27, No. 10, Oct 1984.

[7] Y. Liqun, M.M. Tanik, D.Y.Y. Yun, "Automated Help with Perspectives in a Rapid Prototyping Environment," working paper, Southern Methodist University, 1987.

[8] G. Lapalme, P. Chartray, "An Educational System for the Study Tasking in Ada," IEEE Transactions on Education, Vol. 30, No. 3, August 1987.

Appendix

The objective of Lab 13 is to get the student acquainted with the following:

[1] Be able to declare and use FLOAT AND BOOLEAN variables.

[2] Understand the assignment statement.

[3] Learn how to convert the value in a variable of type INTEGER into a value of type FLOAT.

[4] Learn the arithmetic operators REM and MOD, and their characteristics.

[5] Be able to instantiate a new FLOAT_IO package for the predefined type FLOAT, and input and output FLOAT's in a readable form using field widths.

[6] Use some control structures such as the IF statement.

Lesson 13 Program :

```
with TEXT_IO, CS100;
procedure LAB_13 is
use TEXT_IO, CS100;

-MOD for VERDIX
use FLOAT_IO;
-package REAL_IO is new FLOAT_IO ( FLOAT );
-use REAL_IO;

-main procedure variables

 TEST_SCORE, MAX_SCORE, PERCENTAGE : float;
 CONTINUE_CALC : boolean;
```

```
-                                                      -
- Procedure READ_SCORES -reads in a maximum possible-
- score and actual test score. Checks for validity. -
-  Tasks:                                              -
-    1. Prompt and read the maximum possible score. -
-    2. Prompt and read the actual test score.         -
-    3. If actual test score not between 0 and max   -
-        -Print out the test score                     -
-        -Print out an error message                   -
-        -Set a stop flag to be passed back to         -
-         invoking procedure.                          -
-    4. Pass the max and test score, along with the -
-       stop flag back to the invoking procedure.    -
```

```
 procedure   READ_SCORES  ( ACTUAL_SCORE, MAX_SCORE: out float;
                             CONTINUE_CALC : out boolean ) is
 MAX, TEST_SCORE : float;
 begin - READ_SCORES

  CONTINUE_CALC := TRUE;
  put   (standard_output,"Please   input  the  maximum   possible

                       score on the test:");
  get (standard_input, MAX );
  MAX_SCORE := MAX;
  new_line (standard_output);
  put  (standard_output,"Please  input  the  student's  score  on
                       this test:");
  get (standard_input, TEST_SCORE);
  if (TEST_SCORE > MAX ) OR (TEST_SCORE < 0.0) then
    put (standard_output,"The score you input");
    put (standard_output, TEST_SCORE, 6, 2, 0);
    put (standard_output,"was not in the range of 0 to ");
    put (standard_output, MAX, 6, 2, 0);
    CONTINUE_CALC := FALSE;
  end if;
 ACTUAL_SCORE := TEST_SCORE;

end READ_SCORES;
```

```
-                                                      -
-  Procedure FIND_PERCENTAGE- finds the percentage.  -
-                                                      -
```

```
 procedure  FIND_PERCENTAGE(TEST_SCORE,MAX_SCORE:in  float;

                             PERCENTAGE : out float ) is
```

```
begin - FIND_PERCENTAGE

 PERCENTAGE := (TEST_SCORE / MAX_SCORE) * 100.0;

end FIND_PERCENTAGE;

-----------------------------------------------------
-                                                   -
- Procedure PRINT_INFORMATION- prints the max score,-
-  test score, and percentage. Then finds the letter-
-  grade and prints this out.                       -
-                                                   -
-----------------------------------------------------

procedure    PRINT_INFORMATION   ( TEST_SCORE,MAX_SCORE,

                                   PERCENTAGE : in float) is

 begin - PRINT_INFORMATION

  new_line (standard_output);
  put_line  (standard_output,"Max Score  Test  Score   Percentage
            Grade");
  put_line  (standard_output,"---------  -----------   ----------
            -----");
  put (standard_output, MAX_SCORE, 6, 1, 0);
  put (standard_output, TEST_SCORE, 11, 1, 0);
  put (standard_output, PERCENTAGE, 13, 2, 0):
  put (standard_output,"        ");
  if PERCENTAGE >= 90.0 then
     put (standard_output,"A");
  elsif PERCENTAGE >= 80.0 then
     put (standard_output,"B");
  elsif PERCENTAGE >= 70.0 then
     put (standard_output,"C");
  elsif PERCENTAGE >= 60.0 then
     put (standard_output,"D");
  else
     put (standard_output,"F");
 end if;
 new_line (standard_output);
 new_line (standard_output);
 new_line (standard_output);

end PRINT_INFORMATION;

-----------------------------------------------------
-                                                   -
-         LAB_13 - Main Execution Block             -
-                                                   -
-----------------------------------------------------

begin --LAB_13

  PRINT_COVER_SHEET;
  READ_SCORES ( TEST_SCORE, MAX_SCORE, CONTINUE_CALC);
```

```
   CONTINUE_CALC then
  FIND_PERCENTAGE ( TEST_SCORE, MAX_SCORE, PERCENTAGE);
  PRINT_INFORMATION ( TEST_SCORE, MAX_SCORE, PERCENTAGE);
end if;

end LAB_13
```

This program reads in a maximum possible score on a test and then an actual test score. It will check the score to insure it is a valid score between 0 and the max possible score. If the score is invalid, an appropriate error message will be given. If the score is valid, then the program calculates the test percentage and print out the raw score, the maximum possible points, the percentage, and a grade. The grade will be based on the table below:

Percentage	Grade
>=90	A
<90 and >=80	B
<80 and >=70	C
<70 and >=60	D
<60	F

LPRINTER: AN EXPERT SYSTEM FOR PRINTER MALFUNCTIONS

Judith H. Merrel
Center for Expert Systems
Department of Mathematical Sciences
Butler University
4600 Sunset Avenue
Indianapolis, Indiana 46208
(317)-283-9723

ABSTRACT

Under the auspices of the Butler University Center for Expert Systems, a knowledge-based system to trouble-shoot local printer malfunctions was developed under contract with Eli Lilly, Inc., an Indianapolis-based pharmaceutical corporation. The expert system, entitled LPRINTER, was developed on an IBM-compatible microcomputer by a team composed of undergraduate students under the supervision of the author. The paper describes the context of the project, including a brief description of the Center for Expert Systems, the initial prototyping phase and the expansion and testing phases. Final conclusions are drawn regarding the educational value to the students and the ramifications for the company. In addition, the system itself is described, together with some of the problems encountered along the way.

INTRODUCTION

Butler University is a small (4000 students), primarily undergraduate, liberal arts university with strong professional and preprofessional curricula. Butler is located in Indianapolis, a revitalized midwestern city of 800,000. In January of 1986, the Butler University Center for Expert Systems was created as a forward-looking response to the emerging technology of applied artificial intelligence. A seed grant from the University was used to purchase various types of microcomputing equipment and artificial intelligence/expert systems software. In addition, the Center has access to the University's VAX 11/780 computer. The Center is directed by the author, who is an Associate Professor of Mathematics and Computer Science, and staffed by undergraduate student interns who are majoring in computer science or computer information systems. The Center provides not only a laboratory for faculty research in artificial intelligence and its applications, but also an "advise and consult" center for central Indiana businesses and industries which want to gain a competitive advantage by applying the technology of expert systems to their businesses. In addition, the Center enables students to engage in "real-world" applications of many of the abstractions studied in the classroom. They have the opportunity to tackle large systems too unwieldy for class assignments and to experience working with "real" users. Such opportunities are rather infrequent for undergraduate students at an institution with no graduate program in the computing sciences.

Among the projects which have been completed by Center staff members is LPRINTER, a diagnosis/prescription expert system for malfunctioning local printers. The system was written under contract with Eli Lilly, Inc., an Indianapolis-based pharmaceutical corporation, and was designed and implemented by undergraduate students together with faculty. This paper describes the initial contacts with the company, the prototype, the problems encountered (along with a few solutions), the final system with its implementation, ramifications for the company, and the educational benefits to the students in using the Center's internship program as a vehicle for growth and learning.

THE INITIAL PHASE

Eli Lilly, Inc. is an international corporation with computer installations of various sorts around the globe. When a Lilly employee encounters difficulty with a particular computer system at any Lilly site in the world, a call is often made to a single Help Desk, located at the corporate headquarters in Indianapolis. The phone is answered by an employee in Computer Operations, who then attempts to assist the person with the problem. Typically, the employees at the Help Desk are not trained computer experts and are generally rotated through various jobs within the company on a six-month basis. Thus they are rarely very knowledgeable regarding the myriad arrangements of Lilly computer systems, networks, and operating environments. Furthermore, computer configurations within Lilly are subject to fairly rapid change.

In August of 1986 Center personnel were approached by Mr. Robert Sweet, a systems analyst at the corporate headquarters of Eli Lilly in Indianapolis. He had read about the Center for Expert Systems in a feature article in a weekly Indianapolis business magazine. Mr. Sweet was intrigued by the idea of expert systems and was looking for a way to interest higher-level administrators at Lilly in the idea. (See [1], for example, for a general introduction to expert systems.) After several discussions Center personnel agreed to write a prototype expert system to assist the employees at the Help Desk. The problem then became one of scale --what focus should our system have in order to allow rapid completion of the prototype?

THE PROTOTYPE

Mr. Sweet and the author then began to keep a log of calls to the Help Desk. After five days of monitoring, it was discovered that while the calls were quite varied, a substantial percentage of them (about 15 per cent) involved the complaint, "I sent something to the printer and it won't print." Thus, it was decided that an expert system to assist in troubleshooting local printer problems would be of benefit to the company and yet would be amenable to quick and relatively inexpensive prototyping. Not only could the eventual expert system aid in solving the printer problems, but it would also prove useful as a means of training and teaching the Help Desk employees about Lilly computer systems throughout the company, thus rendering them more efficient in answering future calls.

The initial proposal to Lilly allotted two months for the prototype and allowed for work by three undergraduate students and supervision by two faculty members. Since Help Desk employees have an IBM-AT personal computer at their disposal and since initial brainstorming had led to the idea of a rule-based expert system, it was decided to write the system in the expert system PC-based shell Insight 2+ [2]. The three students were already familiar with this shell, having worked as research assistants in the Center during the summer of 1986 under the auspices of Butler University student/faculty research grants. Each had written at least one modest expert system in Insight 2+. Mr. Sweet agreed to serve as a resource and as the domain expert on the proposed project.

The prototype, consisting of a production rule system with 18 rules, was completed in about six weeks. This system prompted the Help Desk employee to ask a series of diagnostic questions involving the network, attachment status, hardware checks, routing, and proper commands. The design of the prototype was based solely on discussions and interviews with Mr. Sweet. The students practiced their interviewing skills, and gained confidence because Mr. Sweet is not only an expert in the domain area but also an experienced systems analyst. In addition, it was between semesters, and therefore the students and faculty had plenty of time to devote to the design. Thus, the prototyping phase of the project went smoothly and quickly with no major obstacles.

EXPANSION OF THE SYSTEM

After completion of the prototype, sufficient enthusiasm for and interest in the project had been generated to warrant expansion to a more extensive system. Lilly approved a contract for $2000.00 to provide hourly wages for students working on the project. Students and faculty were granted contractor's badges so that they could have free access to Lilly corporate headquarters without a company escort. We were also supplied with an office, a telephone, and a PC on site for our use. In addition, Lilly purchased the Insight 2+ software which is necessary to run the system.

In order to gather more information about the procedures being used at the Help Desk, students and faculty began to observe the operation first-hand. During the first few visits, we were introduced to personnel at the Help Desk. In particular, we met the employee who was currently handling the distress calls, Mr. Rex Walker, as well as other employees in the Computer Operations and Systems Divisions. The students installed Insight 2+ and the prototype at the site. (In fact, the students located and corrected a bug in the installation program supplied by the manufacturer.) We then solicited reactions from Mr. Walker. At this point, we had our first encounter with what would prove to be a substantial problem throughout the project. The points of view of Mr. Sweet and Mr. Walker were radically different. Mr. Sweet viewed the entire enterprise with a systems analyst's "slant," whereas Mr. Walker maintained a much more narrow focus. As we talked with each of them individually about the system and its expansion, we tended to receive conflicting opinions about how things should work. The problem of resolving conflicting opinions from two domain experts is well-known to knowledge engineers and is called the problem of knowledge fusion. Mr. Sweet's experience and training had led him into a broader-based framework in which to view the project, but Mr. Walker, who was actually going to be using the system, had been at the Help Desk long enough to have learned something about diagnosing printer problems. This problem plagued us throughout the construction of the system and led to multiple changes in design. This backtracking might have been avoided had we insisted on involving Help Desk personnel from the beginning of the prototyping phase.

From reading about the development and implementation of other expert systems projects [1], we had anticipated some degree of hostility or anxiety among the eventual users of the system. This never surfaced. One reason may be because of the enthusiasm and ingenuousness of undergraduate students. The students were never perceived as a threat and the entire project was enthusiastically supported by Operations personnel from the outset.

After the expanded system was up and running, the students and faculty involved made a presentation to the upper-level management at Eli Lilly. The students acquitted themselves quite well despite extreme nervousness. The presentation involved overhead slides, as well as a demonstration of the system, and the audience included two vice-presidents and three division managers at Lilly. The presentation was planned and delivered almost entirely by the students, except for some brief introductory remarks by the author. It was quite a heady experience for both the students and the faculty!

THE SYSTEM

In this section we present an informal description of the system itself. LPRINTER is a rule-based, backward-chaining expert system [1]. It attempts to prompt Help Desk personnel so that they may determine what to suggest in order to get a printer which is attached to the correct software application and/or system to produce the desired output. If this fails to correct the problem, LPRINTER provides, through access to dBase-III-type files, the name and phone number of an appropriate Lilly individual or group to contact for assistance. On any path through the system, LPRINTER may instruct Help Desk personnel to type commands on one of several system-dependent terminals located at the Help Desk, to ask the caller pertinent questions, or to call a contact for further aid. The program is terminated when either the printer begins functioning or the system prompts the user to call a contact from an appropriate area. The program may also terminate by virtue of being unable to reach a conclusion because of the lack of necessary technical information such as the System Code or the Transaction Code.

Like all expert systems developed with Insight 2+, LPRINTER operates by "firing rules" (essentially IF-THEN statements) until a goal is reached--in this case, there is only one goal: THE CONSULTATION IS OVER. The system includes the knowledge base consisting of about 75 rules, the inference engine (supplied by Insight 2+), four Pascal subroutines, and two databases. The Pascal programs implement searches of the databases. Two of the Pascal programs are used when the system determines that an outside contact must be made. These programs search the appropriate database (depending upon previous information) and produce the name and Lilly phone number of a first contact, a second contact, and a last resort contact. The other two Pascal routines are used to assist the Help Desk employee in determining the correct codes and identification for the application and/or system with which the caller is having trouble. This information is vital in determining why something won't print and is often not known by either the caller or the Help Desk employee. The required code is then returned to the knowledge base and used to decide what to ask or try next. If the system user is not able to provide enough information so that the system can determine these codes, the session ends and the program terminates with an appropriate message.

A consultation with the system is initiated by the Help Desk employee. LPRINTER responds by requesting preliminary information regarding the system and the network status of the particular printer in question. If, for example, the printer is not attached to the appropriate network, the system then uses a Pascal subroutine to access the names of the contacts in the telecommunications division of Lilly. After gathering the initial information, the system proceeds to determine the attachment status of the printer and branches to an appropriate path depending upon the determination. Other various prompts are made throughout the system, such as requesting a user hardware check (e.g., is it plugged in?).

CONCLUSIONS

In a liberal-arts-based, primarily undergraduate, institution, it is difficult to discover ways to integrate student research experiences into the computing sciences curriculum. Projects such as LPRINTER, developed under an internship program with the Center for Expert Systems, are excellent tools for facilitating such research. The students involved in this project not only learned the "nuts and bolts" of designing and coding an expert system using a shell, but they also experienced firsthand the entire system development life cycle. They were able to see for themselves problems and situations that had previously been only mentioned in texts or discussed in classes. Such experiences have proven invaluable, both for students going on to graduate school in computer science and for those entering the job market. Most graduate programs in computer science now require at least some undergraduate research experience, which is often difficult to arrange at a school such as Butler. In addition, many Fortune 1000 companies, as well as some smaller businesses, are seeking beginning-level employees with some knowledge engineering experience. One of the students who worked on LPRINTER was hired by a large software house, primarily because of the expert systems experience gained on the Lilly project.

The original motivation of Mr. Sweet was to "hook" Lilly executives on the potential benefits of applied artificial intelligence. That goal has been realized. Based on the results of our project, Lilly has invested company resources into an artificial intelligence project with ramifications throughout the company. Discussions are ongoing in such diverse areas within Lilly as toxicology, drug synthesis, and statistical support. These developments are gratifying to Center personnel, since serving central Indiana businesses and industries was one of the original purposes of the Center.

BIBLIOGRAPHY

1. Harmon, Paul, and King, David. *Expert Systems: Artificial Intelligence in Business.* New York, NY: John Wiley and Sons, Inc. 1985.
2. Insight 2+, Level Five Research, Indialantic, Florida, 1986.

AN INTRODUCTORY COMPUTER SCIENCE COURSE USING LOGO: A CASE STUDY

Ernie Giangrande Jr. and Peter Allmaker
Computer Science Department
North Adams State College
North Adams, Ma. 01247
(413) 664-4511 ext. 589/(802) 694-1310

ABSTRACT

The faculty in the Computer Science Department at North Adams State College recently decided to change the programming language used to support CS-151, the first course in the computer science curriculum, from Pascal to LOGO. The history of CS-151 suggests that there has been an ongoing attempt to adapt this course to the state of the art in this discipline and to the needs of the students enrolled in it. Changing languages at this time is viewed by the faculty as another logical step in the evolution of the course. This paper describes the context in which this decision was made, the process used to evaluate LOGO's appropriateness as an introductory language, and the rational upon which the decision was made.

I. INTRODUCTION

The faculty in the Computer Science Department at North Adams State College (N.A.S.C.) have recently decided to change the programming language used to support CS-151, the first programming course in the computer science curriculum, from Pascal to LOGO. This change is viewed by the faculty as another logical step in the evolution of the course.

Changes in the course CS-151 at N.A.S.C. over the past decade and a half have been an ongoing attempt to meet the needs of the students enrolled in this course, improve the teaching of this course, and to incorporate into it contemporary computing concepts and techniques. These changes can be viewed as part of a natural evolutionary process which is continuing today. Those of us teaching CS-151 are constantly looking for more effective ways to meet the goals we have set out for this course.

The purpose of this paper is to examine the decision to change the language used in this light. First, to set the stage for discussing this decision the context in which it was made will be described. A brief review of this course's history including the hardware and software environment in which it has been offered will be presented. Next, a description of the changes in CS-151 leading up to this decision will be given in which we consider both the structure of CS-151 and the students it serves. Our view of this course will be considered next including the objectives for the course and topics to be covered. Then, Pascal and LOGO will be evaluated as languages for supporting this course. Finally, a rational for deciding to switch to LOGO in CS-151 will be presented along with an informal evaluation of that decision.

II. HISTORICAL CONTEXT

It will be instructive to review the historical events which have brought us to the current situation in CS-151 before considering the details of this course and our decision to use LOGO to support it. This will be useful if for no other reason than to demonstrate that this course has gone through a steady stream of changes. We sometimes forget that currently accepted wisdom came about after a number of earlier experiments. Viewed as part of a history of change, a new idea may be less intimidating.

The historical context will be considered from three perspectives: (a) the hardware used to support CS-151; (b) the software used to support CS-151; and (c) the evolution of CS-151 given the students serviced by this course at N.A.S.C. Each of these will be addressed in turn in this section.

Hardware Resources

The earliest offering of what has become known as CS-151 was supported by a mainframe computer accessed remotely via an RJE station at the college. The computer ran under a batch operating system. Gradually the original cards and keypunch machines were replaced by teletypewriter devices. These devices were much more convenient than punching cards but still left much to be desired. Finally, Video Display Terminals (VDT) started to become available. VDT devices are widely available today, as are a variety of microcomputers which are as powerful as many of the early mainframes. These microcomputers can be used as stand alone computers or, by using terminal emulation software, they can be used to access a mainframe computer.

Software Resources

Systems software has evolved in conjunction with the computing hardware. VDTs and other types of terminals are supported by a time sharing operating system which allows many users to be accessing the computer with what they perceive to be concurrent access. Other support software has also became available to on-line users including powerful line and screen editors, program development and program maintenance tools, and a variety of utilities in program libraries. Several programming languages also became available over time. It is the programming languages which are of primary interest to this discussion.

During the early to mid seventies most computing courses were supported by languages like FORTRAN, COBOL, and BASIC. These languages served well for many years. They were widely available and there were a number of text books written for students at the introductory level. FORTRAN and COBOL are still used extensively in industry attesting to the popularity of these languages and the number of programmers trained in their use.

Around the mid seventies Pascal was introduced and has been a powerful force in academic computing ever since.

Much of Pascal's popularity can be attributed to its clean design and the ease of porting it to a variety of computer systems. In acknowledgment of these strengths, the ADA programming language design team used Pascal as a starting point for their work. While Pascal is also steadily gaining direct and indirect support in industry, education remains its stronghold.

Evolution of CS-151

The Computer Science Department at N.A.S.C., like many Computer Science Departments, began as an offshoot of the Mathematics Department. At first a few courses were offered as part of the mathematics program. Eventually a separate department was formed which initially offered only a minor in computer science.

Originally, CS-151 taught programming in FORTRAN and was designed to service two groups of students. The first consisted of those students interested in completing a minor in computer science and who would be taking additional courses in computer science. For these students CS-151 laid the foundation for future computing courses. The second group consisted of students interested in only being exposed to computing and who would likely not take additional courses. This contrast in interest became more pronounced following two events at the college. First, CS-151 was included in the general education curriculum as an option in the Math/Logic category. A n increasing number of students began to opt for CS-151 to satisfy this requirement. Second, the department began to offer a major in computer science in 1979 which attracted a large number of students.

The desire to meet the needs of both groups led to CS-151 being separated into two courses, one for majors and minors, and one for all other students. Both of these courses satisfied the general education requirement in Math/Logic. To differentiate between these two courses the former will be referred to as CS-151 and the latter will be referred to as GE-151. CS-151 remained much the same as it had prior to the separation. GE-151 was primarily a computing survey course. It did, however, have a formal lab component which introduced BASIC programming.

Shortly thereafter the faculty decided that a course in the Math/Logic category of the general education curriculum should be more than simply a survey course and it was once again changed. GE-151 became primarily a course in BASIC programming. At about the same time Pascal was selected to replace FORTRAN as the programming language for CS-151.

After several years it was decided to drop GE-151 returning to the earlier situation of having only one introductory computing course, CS-151. This decision was made because of a perceived difference in the rigor between these two courses. Both groups of students were to be serviced by this one course using Pascal.

This situation remained static for a few years. Then in the fall of 1986 it was decided to again change the language used in CS-151. LOGO was selected to support this course. The rationale for this decision will be presented in section IV.

With this historical perspective we can now turn our attention to the structure of CS-151, and how Pascal and LOGO meet the goals and objectives for this course.

III. CONSIDERATION OF CS-151

In this section three issues will be addressed. First, a brief review of CS-151 will be presented including the objectives for this course and the topics to be covered. Then Pascal will be considered to determine how it meets the stated goals for this course. Finally, LOGO will be evaluated to see how well it meets these same goals.

N.A.S.C. has based much of its computer science program on the guidelines presented in Curriculum '78 [2]. The guidelines for CS-151 will be briefly reviewed here.

Curriculum '78 lists three specific objectives for the first course in a computer science curriculum as follows:

a) to introduce problem solving methods and algorithm development,
b) to teach a high level programming language that is widely used, and
c) to teach how to design, code, debug, and document programs using techniques of good programming style.

This document goes further by identifying topics to be covered in this first course. The topics fall into two primary areas:

a) Programming Language and Programming
  - data types
  - constants
  - variables
  - expressions (arithmetic and logical)
  - control structures (sequencing, iteration, and alternation
  - assignment statement
  - arrays
  - subprograms and parameters
  - simple i/o

b) Algorithm Development
  - problem solving techniques
  - flowcharting
  - stepwise refinement
  - simple numerical examples
  - algorithms
    - sorting
    - searching

These two topic areas account for ninety percent of the course content specified for an introductory course.

While CS-151 at N.A.S.C. follows these guidelines fairly closely, there is one point which should be clarified: Curriculum '78 lists the three objectives for this course and one can assume that they are listed in the order of importance from the proposer's point of view. We have taken the differing view that objectives a) problem solving and b) the programming process, in that order, are of primary importance and that teaching a particular programming language is the least important of these objectives. This is consistent with the point of view put forward by Abelson et. al. that "...the essential material to be addressed by a subject at this level is not the syntax of particular programming language constructs,..., but the techniques used to control the intellectual complexity of large software systems." [1]. We take this to mean that the problem solving techniques and the program design concepts taught in this course are of primary importance.

It is obviously important to teach a programming language in this course but we feel that the language should be viewed as a tool for practicing the concepts being introduced in the course. Therefore, any language which can support the objectives and the topics listed above would be appropriate for use in this first programming course. We will consider two languages in the remainder of this section to see how they map to these objectives and topics.

Pascal, the first language to be considered, has been the most influential programming language in computer science education for the past decade. It was proposed as an instructional language to aid in the teaching of structured programming concepts. Pascal supports a variety of built-in data types including arrays. It also supports variables and user defined constants. Arithmetic and logical expressions are provided in Pascal as well as the prescribed control structures (sequencing, alternation, and iteration). Procedures and functions are supported and both can be passed parameters. Input and output are simple and straight forward. In terms of the programming language constructs required, Pascal is a suitable language to support CS-151.

The objective to teach problem solving and algorithm design, as well as the topics listed under algorithm development, are not dependent on any particular programming language, however, some languages provide support for these concepts better than others. Pascal lends itself to many of these concepts in that it can be used to parallel many of the topics listed above under Algorithm Development. For example, the ability to code procedure and function stubs provides a mechanism for stepwise implementation of programs which parallels the concept of stepwise refinement of a problem solution.

From this it should be clear that Pascal is a satisfactory language for use in the first programming course in a computer science curriculum. Logo will now be evaluated in the same way to determine its suitability for use in this course.

There are a variety of LOGO implementations but in the discussion which follows it is assumed that we are considering the implementation by IBM running on an IBM PC or compatible [3]. For those unfamiliar with this implementation it should be noted that an extensive list processing component is included in the language along with the more familiar 'turtle graphics' features. The list processing is similar to that supported in many implementations of LISP.

LOGO supports a variety of data types including lists. It does not provide arrays specifically but these can be implemented as lists. Variables are supported but not constants, with variables using dynamic typing (types are determined at run time). LOGO provides both arithmetic and logical expressions. The arithmetic expressions can be represented in either infix notation or prefix notation. The primary control structure is recursion but iteration can be used (though not in a way otherwise consistent with the nature of the language). Simple alternation is supported, that is, an IF statement is supported but there is no CASE statement for more complex selection. Obviously nested IF statements can be used to support complex selection. Procedures and functions are supported in LOGO and both can be passed parameters. Non-local reference scoping is dynamic, that is, follows the calling sequence used in a program. If strict modularization is used this is of little concern to the programmer. LOGO supports simple but straight forward input and output constructs. In terms of the programming language constructs provided in LOGO, it too is a suitable language to support CS-151.

LOGO provides features which can support many of the topics listed under Algorithm Development. As noted above, these topics do not depend on the direct support of the programming language being used but features in a language can lend themselves to aiding in the design of a software system. Subroutine stubs can parallel the stepwise implementation of a problem solution in LOGO. Interpretation is used to run LOGO programs which provides the ability to develop and test low level modules easily. These modules can then be incorporated intact into larger programs.

From this it should be clear that, as with Pascal, LOGO is a satisfactory language for use in the first programming course in a computer science curriculum.

There are also implementation aspects of these languages which were considered in deciding on an appropriate language for use in CS-151. These aspects will be addressed in the next section.

IV. CS-151 AT N.A.S.C.

During the 1985-86 academic year the faculty in the Computer Science department re-evaluated CS-151 and considered this course in terms of its objectives and the students serviced. It was noted that many students were having difficulty grasping the concepts being introduced. Most of the students having difficulty were in the group of non-majors who were using this course to satisfy their Math/Logic requirement. We also considered the objective to introduce problem solving techniques in this course and our success in doing so. It became obvious that we were not meeting the needs of some of our students and that we were not meeting the objective to teach problem solving. Each of these issues will be expanded below along with a rational for our decision to switch to LOGO as a solution to these problems.

Let us first consider the students in this course. Many of these students are not highly motivated in terms of computing. They want to accomplish two things by taking this course. First, they want to satisfy a general education requirement. Second, they want a minimal exposure to computing so that they can either include it on their resumes or get some introduction to courses in their own majors which use computers in some way. We turned our attention to the motivation of these students and considered why it was so

If we consider Pascal at this introductory level two things become clear. First, there is a lot of 'intellectual overhead' one must support before a simple program can be developed and implemented. There is the structure of the language (its syntactic rules for forming an acceptable program). The need to compile programs and what that means must also be understood to a limited degree before a program can be run. In addition, students must learn to use a separate text editor to enter their programs initially. For those students interested in majoring and minoring in this area there is a degree of motivation to learn these ancillary concepts if we assure them that it is important to do so. But for many of the other students in the class this seems like a waste of time.

Secondly, for a course that is designed to introduce problem solving techniques the problems studied tend to appear trivial. The power of the techniques we attempt to teach begins to be appreciated only when the problems become large and this is usually not the case in an introductory course using Pascal. Again, students majoring or minoring in this field can be encouraged to learn and apply these techniques because they are told that they will become more important as they work their way through the rest of the curriculum. Non-majors will not likely take additional courses and therefore are not motivated to embrace these concepts.

What we wanted was a language that would meet a number of our needs and those of our students. The following is a list of the criteria we identified for this programming language:

a) it should require the students to consider as little overhead as possible before they can run their first program,
b) it should allow for the development of large programs which can be used to demonstrate the potential for the problem solving techniques introduced in class, and
c) it must support the constructs considered necessary in a course designed around the guidelines in Curriculum '78.

LOGO seemed to be a satisfactory solution to our problem. It meets the first criterion since it provides students with the ability to begin 'writing' programs on the first day of class using LOGO's turtle graphics in interactive mode. Shortly thereafter they can begin to become familiar with the built-in editor which uses many of the editing commands available to edit instructions in interactive mode. Since LOGO is an interpreted language there is no need to consider compiling and all that compilation implies. In addition, turtle graphics provides a fun environment in which to become comfortable with the system and yet some fairly interesting issues can be considered in this context.

The second criterion is that large programs can be developed in this first course. We considered some generally interesting problems and their solutions in both LOGO and Pascal (e.g. a simple data retrieval system for telephone numbers). The solution and implementation of such a problem could be completed in one semester using LOGO but would require two or three semesters of background for students to complete it in Pascal.

The third criterion really addressed the needs of those students majoring or minoring in computer science. We wanted a language that would meet the first two criteria yet not at the expense of those concepts and constructs we felt were important foundations for additional course work in the computer science program. LOGO provides the continuity we sought. Procedures and functions which can take parameters provide support for the structured, modular programming style we wished to encourage. The use of prefix and infix notation for arithmetic expressions is consistent with topics we introduce in the second programming course. Recursion is introduced in a very natural context to solve a variety of problems. The concept of lists and list processing, which we devote a great deal of attention to in CS-151, is a very powerful idea and we have an opportunity to introduce it early in our students careers.

From this evaluation of LOGO in terms of the criteria listed above it was decided that LOGO was a reasonable choice if we were to satisfy the needs of all of the students enrolled in CS-151 and yet still remain consistent with the guidelines of Curriculum '78 for the first course in a computer science program.

V. CLOSING COMMENTS

We have been using LOGO for about a year now and informal evaluation of CS-151 suggests that this experiment is satisfying the criteria we set for this course. Students are on the computer the first day of class and are developing interesting graphics in a very short time. Understanding the concept of programs and the use of the system editor to create programs seems to be fairly straight forward. Over the course of a semester problems of considerable size are considered and their solutions are implemented by the students. For example, in the fall semester of 1986 students developed a menu driven system to maintain a telephone directory which allowed users to add, delete, lookup and modify entries in the directory. This project was developed in a stepwise fashion and consisted of a large number of modules along with lists nested to many levels. Recursion was obviously used extensively but was introduced in a very natural way given the structure of the data to be manipulated.

Though the faculty will continue to monitor this course it seems, at least for now, that we have found a way to meet all of the goals we identified for CS-151. It will be instructive to do a longitudinal evaluation to see how this affects the preparedness of students for the next programming course, and future courses, in the computer science program.

BIBLIOGRAPHY

[1] Abelson, H., Sussman, G., and Sussman, J., *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, Ma, 1985.

[2] ACM Curriculum Committee on Computer Science, "Curriculum '78: Recommendations for the Undergraduate Program in Computer Science", *Communications of the ACM*, Vol. 22, No. 3, March 1979.

A THEORETICAL EXAMINATION OF COMPUTER PROGRAMMING AND MATHEMATICAL PROBLEM SOLVING

Leah P. McCoy
Indiana University at South Bend

Norman R. Dodl
Virginia Polytechnic Institute and State University

ABSTRACT

Five component problem solving skills (general strategy, planning, logical thinking, algebraic variables, and debugging) were identified as common elements of both computer programming and mathematical problem solving. Based on the similarities of these general skills in specific contexts, a theory was generated that the skills would transfer and that experience in computer programming would cause an improvement in mathematical problem solving.

INTRODUCTION

Computers are an integral part of today's world. Nearly every facet of life involves computers, from the supermarket check-out line to the Internal Revenue Service. Many jobs involve the use of computers in some capacity. Largely because of this widespread use (and the resulting demand from the public), schools are adopting the goal of "computer literacy" for their students.

Because computer education is still at a formative stage, "computer literacy" is not well defined. Some educators favor requiring computer literacy courses including programming for all students (Kulm, 1984; Luehrmann, 1981). Others favor limiting computer literacy to machine operation and utilization of pre- programmed software (Bork, 1985; Lockheed & Mandinach, 1986). In order to use a computer effectively in most software applications, no knowledge of programming is necessary. Only a small percentage of students will pursue careers as professional programmers. So, if we are to justify teaching computer programming to all secondary school students, some more generalizable benefit must be found.

A frequent "justification" is that computer programming improves problem solving ability. For the purposes of this paper "problem solving" is defined as a case in which: a problem is presented for which the problem solver wants a solution; a solution is not readily available; and the problem solver must use existing knowledge in new and different ways in constructing a solution to the problem. The claim that computer programming experience improves problem solving ability is supported by a small amount of empirical research (Foster, 1972; Hart, 1982; Johnson & Harding, 1979; Mayer, Dyck, & Vilberg, 1986; Nowaczyk, 1984;) and also by a number of untested, but logically consistent arguments (Lepper, 1985; Nickerson, 1982; Papert, 1980; Shneiderman, 1985).

Much of the logical justification for the theory that computer programming skills may transfer to other problem solving activities is based on the similarity of the two exercises. The process of writing computer programs can be considered a problem solving activity, because it satisfies the definition of problem solving (Kulm, 1984). Since courses in programming are necessarily activity oriented, they provide extensive practice in problem solving in a specific context. Therefore, it is possible that computer programming may improve general problem solving ability.

Mathematical problem solving is an area of interest because a goal of mathematics education is to prepare students for the future (Gagne, 1983; Kilpatrick, 1969). Since the future is difficult to foresee, students must be taught to apply mathematical skills and concepts to novel situations. The National Council of Teachers of Mathematics (1980) has targeted problem solving as the "focus" of school mathematics of the 1980's. Similarly, the National Council of Supervisors of Mathematics (1977, p. 2) stated that "learning to solve problems is the principal reason for studying mathematics." Nearly every mathematics curriculum K-12 includes some goals aimed at developing mathematics problem solving skill. Even with this attention, students' mathematical problem solving skills are often poor. In a review of the results of the second National Assessment of Educational Progress, Carpenter, Kepner, Lindquist, and Reys (1980) noted the widespread failure of students on items dealing with solving mathematical problems.

In this paper the relationship of mathematical problem solving and computer programming problem solving is examined. A general problem solving model, based on the theories of Polya (1957, 1962, 1965) and of Newell and Simon (1972), is presented. Next, the concept of transfer is discussed. Finally, potentially transferable skills which are critical for problem solving in both mathematics and computer programming are explored.

A PROBLEM SOLVING MODEL

A classic model of problem solving presented by Polya (1957), includes the following steps:

1. Understanding the Problem,
2. Devising a Plan,
3. Carrying Out the Plan, and
4. Looking Back.

Even though the model appears linear, in practice it is recursive. Current and previous steps are often repeated as evaluation occurs throughout the process. This model can be seen as an overall strategy that drives the human information processing system during problem solving. It is stored in long term memory and is accessed when a problem is encountered. It functions as a general framework within which problem solving is carried out. As such, it is a general strategy for solving a problem--whether mathematics, or computer programming, or some other content area. The specifics within the four steps may vary slightly, but the overall model applies to all areas.

Another classic theory of human problem solving (Newell & Simon, 1972) can be used to explain the specific internal processes that occur during problem solving. This model characterizes problem solving as a search through the memory system for potential solutions. This search is defined by the external task environment called the problem and the internal representation called the problem space. The problem solver's long-term memory, which consists of a structured representation of experiences and operations, is the system searched. This structure is hierarchial and is believed to be in the form of conditional productions (Brooks, 1977; Mayer, 1987; Resnick & Ford, 1981). Once a schema is returned from long term memory and evaluated in terms of the problem space in short term memory, possible solutions and/or methods are ranked and the most favorable one is chosen. It is then evaluated in terms of the original problem. The process may be repeated recursively until a satisfactory solution is reached.

While these two models may appear different, they are actually quite compatible. The Polya model is a system for the control processor that organizes the problem conditions and sets up the search. The Newell and Simon theory explains the specifics of the internal information processing that occurs as a search is executed. Therefore, the two theories are complementary, with Polya providing the overall strategy, and Newell and Simon explaining the specific internal processes.

TRANSFER

The premise for expecting a positive effect on mathematical problem solving from computer programming is that certain skills and techniques will transfer from one task to the other. Transfer refers to the application of knowledge and skills learned in one task to a different task.

Historically, transfer has been a topic of interest to educators and psychologists. The first 200 years of American education, beginning with the Latin school movement, were primarily based on the approach of faculty psychology (Hudgins, 1977). This is the idea that training in Latin, Greek, and mathematics would train the mind and develop such faculties as memory and reason (Hudgins, 1977; Mayer, 1987). This theory gradually fell out of favor and was replaced by a totally opposite viewpoint, Thorndike's theory of identical elements (Thorndike & Woodworth, 1901). This position is that transfer is possible only if the two tasks contain identical elements. Based on a study of high school curricula, Thorndike (1924) concluded that knowledge and skills were the important result of study of any particular subject, and that no general benefit should be expected.

Osgood's (1949) surface is a classic transfer model which shows that maximum positive transfer occurs when the stimuli and the responses of the original and the new task are identical. Lesser degrees of transfer occur when the tasks are more dissimilar. Hudgins (1977) differentiates between near and remote transfer, depending on the similarity of the original and the new learning task. Several authors agree that near transfer is much more likely (Frederiksen, 1984; Hudgins, 1977; Kilpatrick, 1978).

While computer programming may not fulfill its prophecy of becoming the new Latin school movement (Mayer, 1987) by providing skills that transfer to remote topics, it is likely that its skills transfer to tasks which are similar to programming. Therefore, the similarity of skills used in computer programming and in mathematical problem solving increases the probability of their transfer.

CRITICAL SKILLS FOR PROBLEM SOLVING

Within the integrated problem solving model, five skills have been identified as critical ingredients for success in problem solving. These are general strategy, planning, logical thinking, variables, and debugging. Within the problem solving process, these skills are applicable to problem solving in both mathematics and computer programming. Therefore, it is likely that transfer will occur.

The integrated problem solving model is a representation of a general strategy for problem solving. This general strategy has been shown to be illustrative of the process which drives both mathematical problem solving and computer programming. This control process itself must be considered in the sense of a unit of strategic knowledge in human memory. It is a procedural schema that is accessed whenever a problem is encountered. Just as any other representation in long term memory, this executive control schema must be encoded. It is improved by numerous connections to various problems, which are encoded as it is used.

The more information that is related to the executive control schema, the more useful it will be. Therefore, the more problem solving experience a person has and the more varied it is, the better he will be at solving problems (Kilpatrick, 1978; Mayer, 1976). Since computer programming involves a slightly different implementation of the general process, it provides additional connections to the executive control schema. This practice improves the accessibility of the procedural general process in long term memory and, thus, improves problem solving in general.

Within the problem solving model, planning is an area of major emphasis. In computer programming, planning is an integral part of the activity, often being formally accomplished with a flow chart. The use of a flow chart is also appropriate and desirable for students attempting to solve mathematics problems. In using a flow chart the problem solver must make each part of his proposed solution explicit. The operations, the variables and the order and relationship of each must be clearly specified. This requires that the planning process be an integral part of problem solving (Lepper, 1985; Nickerson, 1982).

Another aspect of planning that is appropriate for both computer programming and problem solving is top-down modular design. Programs are often designed in a structured manner so that the problem is broken down into successively smaller subproblems. The final design is represented with a structure chart, which illustrates the top-down arrangement. In highly structured languages, such as Pascal and Logo, the subproblems are solved with procedures which are combined into modules which are combined with the main program. A similar process in structured BASIC uses subroutines to solve the subproblems. This design strategy is essential in solving large programming problems. While it is not a requirement in all mathematical problem solving, it can be used with large mathematics problems with very positive results. Using subgoals is a problem solving strategy suggested by both Newell and Simon (1972) and Polya (1962).

Development of logical thinking is a desired goal for all students. However, this goal is not well defined and not well achieved. A recent conference of mathematics educators concluded that even though geometry courses are meant to teach logic, they do not achieve that goal (Fey, 1984). Only a few secondary students complete courses in formal logic, which are theoretical and tedious.

Computer programming offers a concrete representation of logical forms. One of the most common structures in many computer languages is the IF...THEN...ELSE... statement. This requires the student to use conditional thinking. In programming, this use of conditionals is extensive and the programmer has experience in concrete representation of this concept (Mayer, 1985). This conditional reasoning is a useful skill in mathematical problem solving.

Since both advanced mathematics and computer programming make extensive use of variables, they both require the student to have skill in understanding and in using variables. Studies have found that many mathematics students have difficulty in understanding variables (Clement, 1982; Clement, Lochhead, & Monk, 1981). Computer programming offers a similar implementation of variables in a more concrete format (Mayer, 1985; Soloway, Lochhead, & Clement, 1982). Programmers must use variables frequently and extensively, and, thus, acquire a familiarity with them that is usually not achieved in a mathematics context.

A final critical skill for problem solving is debugging. There are two types of debugging which are involved in programming. First, any syntactic error in code is detected by the interpreter or the compiler, and an appropriate error message is generated. Thus, the problem solver gets immediate feedback, and many errors are corrected at an early stage. Even though this immediate external feedback is not always present in mathematical problem solving, practice may help the student to acquire the habit of making syntactic checks himself.

The other type of debugging is checking a program which appears to execute properly but gives an incorrect result. This is called a logic error and is a part of the looking back step of problem solving. The unique contribution of programming to this activity is the concrete result which is obtained quickly and easily. This encourages hypothetical thinking, where the student investigates "What if?" questions (Lepper, 1985; Nickerson, 1982). In mathematical problem solving, this endeavor is considerably more time consuming and more often neglected (Paige & Simon, 1966). Performance of this activity in the programming context would make the student more aware of its value, and more willing to incorporate it into mathematical problem solving.

SUMMARY

Both mathematical problem solving and computer programming have been described in the context of an integrated problem solving model. These two activities have been shown to be similar in several ways. It appears that computer programming provides experiences that develop critical problem solving skills which will transfer to mathematical problem solving. These include (1) a general problem solving strategy, (2) planning skills, (3) logical thinking skills, (4) understanding of algebraic variables, and (5) debugging skills. Therefore, experience in computer programming should have a positive effect on mathematics problem solving ability.

REFERENCES

Bork. A. (1985). *Personal computers for education.* New York: Harper and Row.

Brooks, R. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies*, 9, 737-751.

Carpenter, T. P., Corbitt, M. K., Kepner, H. S., Lindquist, M. M., & Reys, R. E. (1980). NAEP note: Problem solving. *Mathematics Teacher*, 73 (9), 427-433.

Clement, J. (1982). Algebra word problem solutions: Thought processes underlying a common misconception. *Journal for Research in Mathematics Education,* 13 (1), 16-30.

Clement, J., Lochhead, J., & Monk, G. S. (1981). Translation difficulties in learning mathematics. *American Mathematical Monthly*, 88, 286-290.

Fey, J. (Ed.) (1984). *Computing and mathematics: Impact on secondary school curricula.* Reston, VA: National Council of Teachers of Mathematics.

Foster, T. E. (1972). The effects of computer program experiences on student problem behaviors in eighth grade mathematics. Doctoral dissertation, University of Wisconsin, Madison.

Frederiksen, N. (1984). Implications of cognitive theory for instruction in problem solving. *Review of Educational Research*, 54 (3), 363-407.

Gagne, R. M. (1983). Some issues in the psychology of mathematics instruction. *Journal for Research in Mathematics Education*, 14 (1), 7-18.

Hart, M. (1982). Using computers to understand mathematics. *Mathematics Teaching*, 52-54.

Hudgins, B. B. (1977). *Learning and thinking: A primer for teachers.* Itasca, IL: Peacock.

Johnson, D. C. & Harding, R. D. (1979). University level computing and mathematical problem-solving ability. *Journal for Research in Mathematics Education,* 10, 37-55.

Kilpatrick, J. (1969). Problem solving in mathematics. *Review of Educational Research*, 39 (4), 523-534.

Kilpatrick, J. (1978). Research on problem-solving in mathematics. *School Science and Mathematics*, 78 (3), 189-192.

Kulm, G. (1984). Mathematics processes and computers in the junior high school. *Mathematics Teacher*, 77, 628-630.

Lepper, M. R. (1985). Microcomputers in education: Motivational and social issuse. *American Psychologist*, 40 (1), 1-18.

Lockheed, M. E., & Mandinach, E. B. (1986). Trends in educational computing: Decreasing interest and the changing focus of instruction. *Educational Researcher*, 15 (5), 21-26.

Luehrmann, A. (1981). Computer literacy--What should it be? *Mathematics Teacher*, 74, 682-686.

Mayer, R. E. (1976). Some conditions of meaningful learning for computer programming: Advance organizers and subject control of frame order. *Journal of Educational Psychology*, 68 (2), 143-150.

Mayer, R. E. (1985). Learning in complex domains: A cognitive analysis of computer programming. *The Psychology of Learning and Motivation*, 19, 89-130.

Mayer, R. E. (1987). The elusive search for teachable aspects of problem solving. In J. A. Glover & R. R. Ronning (Eds.), *Historical foundations of educational psychology*. New York: Plenum Press.

Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: What's the connection? *Communications of the ACM*, 29 (7), 605-610.

National Council of Supervisors of Mathematics. (1977). *Position paper on basic mathematical skills*. Washington, DC: National Institute of Education.

National Council of Teachers of Mathematics. (1980). *An agenda for action*. Reston, VA: NCTM.

Newell, A. & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Nickerson, R. S. (1982). Computer programming as a vehicle for teaching thinking skills. *Thinking: The Journal of Philosophy for Children*, 4, 42-48.

Nowaczyk, R. H. (1984). The relationship of problem-solving and course performance among novice programmers. *International Journal of Man- Machine Studies*, 21, 149-160.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological Review*, 56, 132-143.

Paige, J. M. & Simon, H. A. (1966). Cognitive processes in solving algebra word problems. In B. Kleinmuntz (Ed.) *Problem solving: Research, method, and theory* (pp. 51-119). New York: John Wiley and Sons.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Polya, G. (1957). *How to solve it*. Princeton: Princeton University Press.

Polya, G. (1962). *Mathematical discovery: On understanding, learning, and teaching problem solving (Vol. 1)*. New York: Wiley.

Polya, G. (1965). *Mathematical discovery: On understanding, learning, and teaching problem solving (Vol. 2)*. New York: Wiley.

Resnick, L. B. & Ford, W. W. (1981). *The psychology of mathematics for instruction*. Hillsdale, NJ: Erlbaum.

Shneiderman, B. (1985). When children learn programming: Antecedents, concepts and outcomes. *The Computing Teacher*, 12 (5), 14-17.

Soloway, E., Lochhead, J., and Clement, J. (1982). Does computer programming enhance problem solving ability? Some positive evidence on algebra word problems. In R.J. Seidel, R.E. Anderson, and B. Hunter (Eds.) *Computer literacy: Issues and directions for 1985*. New York: Academic Press.

Thorndike, E. L. (1924). Mental discipline in high school studies. *Journal of Educational Psychology*, 15, 1-22,83-98.

Thorndike, E. L. & Woodworth, R. S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, 8, 247-261, 384-395, 553-564.

STUDENTS AS DEVELOPERS OF EDUCATIONAL COMPUTING PROJECTS: EXPERIENCES FROM CARNEGIE MELLON UNIVERSITY

Carol Scheftic
Center for Design of Educational Computing
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-7644
scheftic@andrew.cmu.edu

ABSTRACT

One of a university's greatest resources is its students. One of Carnegie Mellon University's greatest strengths is the variety of opportunities it makes available to its students. Just two years ago, an undergraduate scholarship program was initiated to take advantage of these features in the area of educational computing. This paper will present examples of several student projects developed over the last two years. It will explain the benefits to the students and to the university community resulting from this program. It will describe how students may join the program by proposing their own projects, and how faculty and staff may be able to get some additional student support for their educational computing projects through the scholarship programs. Finally, it will describe a few areas where our thinking has changed as we become more experienced at managing the program, in the hope that such information will be useful for other groups wishing to develop a similar program.

Despite continuing advances in the power, sophistication, and availability of computing machinery, there exists only a relatively small amount of top-quality software and support materials for educational applications. Programmers, instructional designers, and faculty members are some of the groups that come to mind when the discussion turns to good candidates for people to help remedy this shortfall. One group is frequently overlooked: students. Today's university undergraduates, in particular, are likely to have enough background in computing technology to be able to undertake significant development projects. Being actively involved in the educational process on a daily basis, they are often aware of opportunities for making a real impact on students' lives. Carnegie Mellon University (CMU) has initiated a program to support and encourage students to design and develop educational computing projects.

1. Institutional Context

In 1982, CMU and IBM agreed to create the Information Technology Center (ITC), based at CMU but with staff members from both organizations (Morris, et al, 1986). Its major goal was the design and development of a distributed computing environment for the university; the system was named Andrew, after Andrew Carnegie and Andrew Mellon. The system was to utilize a number of new and powerful technologies, such as the advanced-function workstations being developed by IBM (IBM PC RT), Digital Equipment Corporation (VAXstation II), and Sun Microsystems (Sun 2 and Sun 3). As personal computers became more ubiquitous across campus during Andrew's development, mechanisms were also added for accessing many of its features through machines such as the IBM PC and PS/2 series and the Apple Macintosh line. Although the ITC was chartered to design and deploy this powerful system throughout the university, so that it would be available for a variety of educational and research uses, their task was big enough that they could not be expected to adequately address all such applications.

Thus, in 1983, the Center for Design of Educational Computing (CDEC) was established to help foster an appropriate intellectual environment for people concerned with the application of these advanced technologies to the educational process. It has a variety of mechanisms for providing support to faculty and staff involved in research and development projects, facilitating technology transfer activities, and providing educational service using the most modern technologies.

CDEC helps to support both applications with educational value and tools for enhancing the development of such applications (Trowbridge, 1986). For example, members of the CDEC staff are responsible for the development of the CT[1] authoring language (Sherwood & Sherwood, 1986). They also develop instructional materials, written in CT and other languages and running on both workstations and personal computers. The materials written in CT exercise new features during their development; the use of various languages and systems encourages frequent comparison among them. These development activities provide materials that are usable in their own right as educational tools; they also serve as models of effective instruction (e.g., Trowbridge, Larkin, & Scheftic, 1987).

Faculty and staff, however, are not the only members of the community interested in advancing the field of educational computing. Graduate students undertake projects in that area, in conjunction with an appropriate faculty advisor. There was no reliable mechanism, however, by which undergraduate students could receive support and encouragement for developing their ideas concerning the appropriate use of computers in education. In fact, their plight was often made worse since, in addition to heavy course loads, a large number of them were committed to work-study jobs, sometimes interesting but often of a very routine nature, simply in order to be able to pay their educational bills.

In 1985, the CMU administration made an innovative offer to CDEC. If the group would design and administer a scholarship program that enabled talented undergraduates to pursue educational computing projects, the university would provide funds for such awards. A task force of faculty, staff, and students was formed to design the program. By the end of that year, the first set of plans had been completed and, in January 1986, the CDEC Scholarship Program in Educational Computing (CSPEC) was launched.

---

[1] CT was formerly known as CMU Tutor.

2. CSPEC's Goal

The goal of CSPEC is to develop a well-balanced group of students, representing a variety of talents and interests, from all colleges and majors within the university, who can work individually and together to effectively plan, design, implement, and evaluate projects in the area of educational computing.

In selecting participants, we look for situations where there is a qualified student, a good project idea, and the resources to carry it out. Projects must be of use to the CMU community, innovative in some way, and they must involve both education and computing. Those guidelines are both flexible and restrictive. For example, a project that looked like an innovation from a computer science prospective, but which had no direct application to an educational issue, would not be supported by CSPEC. Neither would an innovative idea, incorporating both education and computing aspects, but utilizing some facet of technology with little to no availability or support on our campus. (To be sure, if we heard of such a proposal, but felt it were not appropriate for support via CSPEC, it is unlikely that we would completely ignore the proposal. Within CDEC, we would look for other sources of support for such an idea.)

As for resources, that is another complex issue. Before agreeing to support a student's project, we try to verify that adequate hardware and underlying software resources are available, that the student has or can obtain the requisite design and development skills, and that there is at least one senior person who will provide supervision and guidance to the project. Some details of how we manage all this are described in Sections 4 and 5.

Since the program is billed as a "scholarship," we do consider students' academic standing in determining their eligibility for an award. We do not, however, have extremely stringent requirements. We are looking for students in good academic standing who also have good ideas, students we feel could benefit from, and provide service to, the program. Thus, grade point average (above passing) is used more as an estimate of the size of the project during a given semester (and the corresponding size of the award) than it is as an absolute measure of eligibility. Achieving this balance is discussed further, in Section 6.

3. CSPEC's Outcomes[1]

Before giving detailed consideration to what are basically management issues for the program, let us first consider some of its accomplishments in just under two years of operation. Of course, not all of the students' projects have been outstanding successes. The ones described below, however, are all successful enough that they are either in actual use already, or far enough along in their development that we feel comfortable in demonstrating them to visitors who want to see the kinds of work done on our campus. In the following paragraphs, the parenthetical notes after a student's name will indicate both major field of study and year of graduation (actual or expected).

It is also worth pointing out that, since the CT language is still in the development stages within CDEC, there are no regular courses in using that language yet available to our students. For those scholarship projects where the language of choice was CT, the students learned to use it as a part of their scholarship responsibilities (a joint testimonial to the ease-of-use of the language and the initiative of our scholars).

3.1. Students can work in their major field of study.

Pamela Reinagel (Biology, 1988) is designing and developing a program that simulates student-proposed experiments in genetic engineering. The goal of this simulation, written in the CT language, is to give students practice in the design of experiments, thus complementing the traditional course focus on how to execute experiments and analyze results. The program uses current models of genetic mechanisms to predict the outcome of a potentially unlimited number of experiments involving a variety of complex sequences of procedures. A student might try out several designs (taking fifteen or twenty minutes each) in order to decide on the one to conduct over a period of time (such as two weeks) in the actual laboratory. This has the double advantage of allowing the student to see the expected outcome of a larger number of designs while, at the same time, making more efficient use of the time conducting live experiments. The student also has the opportunity to design and test experiments far beyond the scope of a typical laboratory course, such as large-scale experiments which would require months of full-time work or advanced experiments which would require hazardous materials not normally authorized for use in student laboratories.

David Thompson (Chemistry, 1988) has designed and developed a program that will aid first year Chemistry students in solving chemical equilibrium calculations. The program, also written in CT, will allow students to understand intricate chemical dynamics and equilibrium concepts without getting lost in the large amounts of algebra traditionally involved in equilibrium calculation. The program was tested with one advanced class in the Fall Semester, 1987; based on the success of that test, it is being used in regular sections in the Spring Semester, 1988. It is publicly available on the Andrew system, and will continue to be available as an integrated part of introductory Chemistry coursework.

---

[1] The author is indebted to CDEC Scholar **Linda Branagan** for her contributions to the CDEC Technical Report (Scheftic & Branagan, 1987) that forms the basis for much of this section.

Theresa Hartsell (Chemistry, 1989) started out by providing Dave Thompson with some assistance on changes to the user interface of his program, as indicated by its first use in actual classes. She did that to gain experience, in both the CT language and the design of user interfaces, and has recently begun work on a program of her own to reinforce students' learning of terminology and notation in organic chemistry.

Christopher Lewis (Biology, 1988) has also designed and developed a program for students from the natural sciences, including biology, chemistry, or biochemistry. Written in CT, it provides an interactive palette on which to draw chemicals. The routine's power comes from the fact that it is very flexible, allowing the students to draw a wide variety of structures of their own choosing. The program also includes a check feature, which remains silent until specifically invoked. At that point, the program goes to a database of "correct" chemicals and compares what was drawn to what the computer considers correct. This is not just a simple matter of comparing two files: the program is able to account for the fact that no two people will draw the same molecule in exactly the same way.

Chris Young (Philosophy, 1990) recently proposed a very different type of project, to conduct an assessment of the use of interactive video as a medium for the presentation of ethical issues. He would work with a team from CDEC who are developing a lesson involving the case of Dax Cowart, a young adult who suffered severe burns and other injuries in an automobile accident. He requested that the medical staff discontinue treatment and permit him to go home to die. Various people involved in the case are interviewed, including Dax himself, his family, friends, doctors, nurses, and counselors; also shown are actual scenes of the painful treatments necessary to keep Dax alive. At repeated intervals throughout the presentation, students are given the opportunity to indicate whether Dax should be allowed to die. Their answers and reasons form the basis for deciding which information the program should present next. The project that Chris has proposed doing with CSPEC support includes collecting protocols of students using the lesson. He would examine them for strengths and weaknesses in the presentation, from both the educational and technical sides. Furthermore, he would consider the program's impact in two disparate settings, with university undergraduates in an ethics course and with both staff and students at a medical school, to determine whether there are differences based on those contexts.

3.2. *Students can, for variety, work in another field.*

Greg Eoyang (Electrical and Computer Engineering, 1987) has worked with Leslie Burkholder of CDEC and the Philosophy Department on an iterated Prisoners' Dilemma program. It was originally written in CT and C, and it allows the user to answer questions about the strengths of various general policies that might be adopted in iterated dilemmas. The program allows the user to change the environment in which the policies or strategies compete. New strategies may be created and tested; they may also be entered in a tournament. Although the problems are generally phrased in terms of alternatives available to prisoners, they investigate the implications of using (or not using) cooperation in various difficult situations, such as negotiations among foreign powers. Although Greg has graduated, his work was so successful that it has been revised so that it is now completely written in CT. This allows it to be easily transportable to individual Macintoshes, as well as being run on the more powerful Andrew workstations. It has also been expanded to allow students (and researchers) to both replicate and extend all extant research on Prisoners' Dilemma problems, enabling students to be directly involved with an on-going research area.

Dan Kahn (Physics, 1988) is working with the English department. He is helping to make the Writer's Workbench^{TM} programs available on the Andrew system. These programs were originally created at Bell Labs to give grammatical and stylistic profiles of a piece of technical writing. Dan is working on a user interface that will make these programs more attractive to the student user.

A number of students have worked on various aspects of another project in the English department, known as WARRANT (Writing and Reasoned Reading About Normative Texts). The purpose of this project is to create a computer environment for teaching critical reading, reasoning, and writing skills.

Keith Evans and Aaron Oppenheimer (both in Electrical and Computer Engineering, 1989) spent a year with WARRANT, where they worked on the design and implementation of (1) a user dialogue for filing and printing preferences, and (2) the collection of statistics to be used in research on cognitive processes in note-taking.

Tina Cacolice (Applied Mathematics / Computer Science, 1988), Leonard Dickens (Applied Mathematics / Computer Science, 1988), and Chuck Lewis (Electrical and Computer Engineering, 1989), as well as Keith Evans and Aaron Oppenheimer (above), have worked on a WARRANT program that uses multiple interfaces to train students to detect and diagnose problems in writing. Users are shown a passage of text, asked to identify words or segments of interest, and allowed to classify the segments according to a number of criteria, both good and bad. The program will record these notations, compare them with an expert's version of the comments, and provide feedback on the results. In this way, students should gain the skills necessary for critiquing their own writing, rather than having to wait for their teachers' comments on their work.

Rich Segal (Applied Mathematics / Computer Science, 1990) came to the rescue when a faculty member visiting our campus for a year from another university suddenly needed some programming assistance. He made the appropriate changes to enable her educational software, designed to be used on individual personal computers, to run more effectively and efficiently on the network configuration used on our campus. Having completed that project to our visitor's satisfaction, Rich is now working on a program for a research project on the uses of interactive computer communications in education. Sponsored by the Center for Educational Computing in English, he is writing an improved version of a program that allows users logged in from any location on the network to have online conversations in real time using special windows open on each person's screen. His enhancements are designed both to improve the user interface and to add new capabilities.

3.3. Students work on all aspects of educational computing, not just programming.

Diane Burton (Social Science, 1987) and Keith Saft (Creative Writing & Graphics Communication Management, 1988) developed a series of educational, multi-media presentations suitable for teaching introductory computing concepts. These presentations have become part of the curriculum for Computing Skills Workshop (CSW), the introductory computing course taught to all incoming freshmen at CMU. This is a particularly interesting course, in that there is one course director and thirty-four (34) undergraduate teaching assistants who are responsible for covering the seventy (70) different hands-on laboratory sections needed each fall. Diane also worked on another project for CSW, developing a week-long training course for the teaching assistants, to be sure both that they had adequate experience on each of the three computer systems they would be introducing to the freshmen and that they had a solid understanding of a variety of teaching strategies and procedures.

Linda Branagan (Professional Writing, 1989) is providing support to various projects on documentation, in the form of both online and printed materials. Her contributions include both reviews of existing documents, such as the textbook used in Computing Skills Workshop, and development of original materials, primarily as a consultant on other scholars' projects. She is also working on a research project within the Information Technology Center, developing guidelines for technical writers on the preparation of instructions for the installation of computer software.

3.4. With experience, students discover ways to diversify their projects.

Wayne Mesard (Psychology, Dec. 1987), who had experience as a teaching assistant with the Computing Skills Workshop, became concerned with the administration of tests in that course. With the large number of sections (70) conducted almost continuously for six days each week, and the close proximity of freshmen from different sections living in the dormitories, having a large variety of examinations was essential. But developing and formatting those exams was a very tedious task, not to mention the difficulty of accurately generating and distributing the corresponding answer keys for the teaching assistants to use. Wayne's solution to this problem was to have the instructor create a database of exam questions and answers, classified by topic and difficulty, and then use XamMaker, a program he wrote, to generate the arbitrary number of different exams and the corresponding answer keys (both accurately formatted). The program has already been used successfully with CSW, and is available on the Andrew system for any faculty member on campus. Wayne is subsequently looking into an extension to this program that will allow the generation, administration, and grading of online exams from the XamMaker files.

Dave Thompson and Wayne Mesard (both mentioned above) have also done some work developing introductory tutorials. These programs are designed to introduce the new user to the Andrew system, where the instruction is provided by completely self-contained programs (no accompanying documentation required, other than the single command needed to start up one of their programs) running on the Andrew system. These are real research projects undertaken by these two scholars, where part of their aim includes producing guidelines for other developers who subsequently attempt similar projects.

3.5. Scholars have the opportunity to apply their skills outside of CSPEC.

During the summer of 1986, the InterUniversity Consortium for Educational Computing (ICEC) held a series of one-week workshops at Carnegie Mellon on the Andrew system, the CT authoring language, and topics in instructional design. Similar workshops have been held at the California State University at Northridge (CSUN) and at Vassar College. ICEC also offers a hotline service for its members on questions related to these topics. Five students who were active in CSPEC during the spring of 1986 (Rajdev Basudev, Gregg LaRue, Pamela Reinagel, Keith Saft, and David Thompson) were hired by ICEC in 1986-87, as teaching assistants for the workshops (at both CMU and CSUN) and as hotline staff, primarily based on the experience they gained through the scholars' program. And when the President of CMU, Richard Cyert, decided that he wanted to learn to write programs in CT, despite the limited time available to a person with his responsibilities, he asked if we had a scholar who might facilitate this process. Dave Thompson, the scholar with the most extensive background in both programming and teaching in CT, has been quite capable in meeting that need.

The entire group of scholars made presentations at the software fair that was hosted by Carnegie Mellon in conjunction with the EDUCOM '86 conference. Four scholars (Roger Aguinaldo, Rajdev Basudev, Diane Burton, and Greg Eoyang) were asked to make a special presentation to the University's Board of Trustees on the ways in which the campus computing projects had affected their lives and their plans for the future. Two scholars (Diane Burton and Pamela Reinagel) were asked to represent the students' view of CMU's computing environment at the official unveiling of the Andrew Network. Several scholars participated in IBM's Advanced Educational Programs conference in Boston in June, 1987. Pam Reinagel's paper, based on her Molecular Genetics simulation, was accepted for presentation and included in the conference proceedings. Dan Kahn and Dave Thompson were invited to demonstrate their programs in the Software Demonstration area. Raj Basudev and Wayne Mesard were invited by the Digital Equipment Corporation to help staff the CT booth at the DECWorld show in Boston in September, 1987. As a result of all of this exposure, a number of the scholars have been interviewed by national and international news media concerning their views on the nature and impact of the various educational computing projects at CMU and their own roles in the development of those projects.

4. CSPEC's Operation

Now that the reader has some understanding of the dimensions of the program, in terms of its accomplishments, the remaining sections of this paper will describe the organizational details that make such progress possible.

4.1. Program Size

Although we had a full year's funding in the 1985-86 academic year, we did not award any scholarships during the fall semester. Instead, we spent that time designing the program and setting up its implementation. For the first round, we were particularly concerned with establishing a common core of terminology, expectations, and experiences for all of the participants. When we initiated the first scholarships in the Spring Semester of 1986, we awarded 20 full scholarships. We also invited five additional students to participate in all our activities as "alternates" - they received no financial award for that semester but we said that, if their work looked promising, they would be given priority for being added to the list of recipients for the next round. Of those five, two declined our offer and three accepted it.

We also required that all students, scholarship recipients and alternates, register for and participate in one or two classes on instructional design (first, one on general instructional design and, then, one specifically oriented toward educational software development). These were offered as part of the University Core Curriculum, and taught by the CSPEC director. Students received up to 12 units (4 credits) for this coursework. While these were very useful in establishing a common vocabulary and sense of community among the students when we started the program, we have been able to assimilate subsequent new members into the group on an individual basis. A few more comments on the courses will be made at the end of this paper, in Section 7.

Since we had awarded no scholarships in that first Fall, we had enough money left to support nine students over the summer. As a result of our experiences in that first semester, we also reconsidered the design of the program. It was at that time that we introduced the possibility of partial awards; that was also when we consciously cut back on the number of awards given during the academic year. Those decisions served to reduce the burden on the program's director and to guarantee that there would continue to be money available to support more intense work on several of the top students during the summer. Table 1 in the Appendix summarizes the size of the program over each segment of the past two years.

To maintain the tax-free status of the academic year awards (see Section 6.1), we are prohibited from advertising the program, nor can we have a formal application process. We do publish articles about successful student projects and we participate in various seminars and conferences where we can highlight different aspects of the program. The scholars themselves tell their classmates about CSPEC, and academic advisors are kept informed of the possibilities. We keep in touch with the people who coordinate computer-related activities on campus, and ask that they keep us informed about promising students whom they encounter. Possible candidates are interviewed individually by the program's director and one or more additional faculty members concerning their interest in the program and in a particular project. The final decision on whether to offer an award takes into account the results of those interviews, the potential value of the project being proposed, and the desire to maintain a group of participants from a variety of majors and with a variety of interests and abilities.

4.2. Supervision

We try to assure that each scholar has both a specific supervisor and several other possible sources of guidance. These people may be selected from the CSPEC Director, other CDEC staff, the person who has agreed to be the project supervisor, other staff within a sponsoring department, or other people that we can identify who seem to have the appropriate skills. We try to have such a support network lined up for the students before they begin work on their projects, although we are prepared to change or supplement the group as new needs arise. Our strategy is to combine the specific guidance of a project supervisor who agrees to help the student, when necessary, to stay on track with the project, with additional resources available across the campus, so that no one person becomes overly burdened with the needs of a student developing an individual project. Table 2 in the Appendix summarizes the distribution over the past two years.

4.3. Intellectual Property Rights

The university and the student sign an intellectual property agreement covering the projects for which scholarships are awarded. In most cases, the university retains "ownership" of all work, but grants the student irrevocable, non-exclusive, perpetual, royalty-free license to use for non-commercial purposes all intellectual property resulting from scholarship project(s). In the few cases where the student has done a significant amount of work on the project as an individual, and only later joins the scholarship program to complete it, then the typical arrangement is to reverse the situation, so that the student does retain ownership of the project but grants CMU a full license to use it. If a situation arose where neither of the above models were appropriate, we would work with the student and the university lawyers to develop a suitable alternative agreement. One main reason for requiring these agreements is to make sure that everyone understands the nature of the relationship from the outset of the project. Acknowledgement of the students' participation in the project development, and of the use of CSPEC funds to support the work, is to be given in all cases.

5. Computing Resources

Carnegie Mellon has an institutional infrastructure that aids the implementation of such a program. This section will provide a brief review of resources that are available to help support CSPEC.

5.1. Hardware

So far, all CSPEC projects have been based on either the Andrew distributed computing environment, using several brands of advanced-function workstations, or personal computers including the Apple Macintosh or the IBM PC series. While some students have their own personal computers and others prefer to work in the public clusters (which are open 24 hours a day, 7 days a week, all across the campus), we do try to assure that the sponsoring department (CDEC or an academic department) has adequate machinery for the scholars to use.

The CDEC staff itself is housed in two different buildings. In each one, we have a room where we make a few computers available on a semi-public basis. That is, our scholars can use them, visitors are invited to use them for short periods of time, and we use them ourselves when visiting our colleagues for discussions and demonstrations. Academic departments often have a similar arrangement, where the primary users may be faculty and graduate students; we ask them to extend the necessary privileges for using equipment to any scholar they sponsor. Since the program is fairly small, this type of arrangement has worked well.

As described above, in Section 3, not all students take on actual programming projects. For the students who developed the multi-media programs for the Computing Skills Workshop, arrangements were made with the Instructional Technology department for training and use of the appropriate audio-visual production equipment, and with Campus Printing for production and distribution of the accompanying papers and bindings. (Once materials are approved for use in a course, any costs incurred are charged directly to the course.)

5.2. Software

Another of the steps in negotiating the design of the project with a potential scholar is making sure that adequate software resources will be available. Again, we've had little trouble in this regard, partly because of the guideline that says a project must be of use to the CMU community - if the underlying software is to be there for later delivery of the project, then it is typically there for development as well.

While we make no requirements that the programming project must be written in CT, it is also true that CDEC is the base for development of that language. Thus, if students choose to work in it, they will be guaranteed to have people around who can help them out with difficulties. Other languages commonly used on our campus, for which support is also readily available, include C, Lisp, and Pascal. There are also word processors, spreadsheets, and statistical programs available for students on both the Andrew network and the personal computers. There is an extensive software library available on campus, where students can review existing, commercially-available software for ideas and comparison.

6. CSPEC's Budget

The program operates on an annual budget[1] of $50,000. During the academic year, there are somewhere between 10 and 15 students involved with the program at any one time. The majority of the awards begin and end with semester boundaries, but there is no requirement that this be so. A few students enter the program during the semester, while others may complete their projects before semester's end.

While a semester is in progress, the maximum award a student can receive is $1,250. For that amount, the students are expected to perform about 200 hours of service to the university on their educational computing project(s). This corresponds to a pay rate of $6.25 per hour, and is the same for all students. It is better than the typical rate earned by students on work-study, although it is not as much as some of the students might make at some of the highly technical programming jobs on campus (particularly those with the Computer Science Department). As described in Section 6.1, these awards are not considered taxable income for the students (or their parents, if the student is still claimed as a dependent), which does make them a little more "competitive." The 200 hour total was picked because it is a "nice, round number," not because it is an exact multiple of anything. The CMU semester is 15 weeks long. Accepting the realities of student life, that the majority of students will devote little time to their projects during the week of mid-terms or the last week before the end of a semester, that gives 13 full weeks to work with. A student putting in 15 hours per week for those 13 weeks would have completed 195 hours in all. The extra five could be put in during that time, or during one of the two weeks when we really didn't expect much work.

Then again, for a student carrying a full course load and participating in several extra-curricular activities, 15 hours each week of project work can be too heavy a load. As a result, we negotiate with all the students the amount of time that they expect they can devote to an educational computing project, and we may give a three-quarter, half, or one-quarter award instead. We prorate the awards based on the maximum so, for example, a student will receive $625 for 100 hours' of service. If students need to take more than one semester to complete their service, that is acceptable as long as they discuss it as soon as possible with the program director. If they take a partial award and find that they are able to complete their service before the semester has ended, we can grant them an additional fractional award to do more work on their project. In fact, we almost always start students with a half-award during their first semester in the program, so that the program director, the project supervisor, and the student have some flexibility in determining the best schedule to follow while minimizing the chance that the student's academic work will be adversely affected by this new commitment.

Since we never seem to use up all the program funds during the school year, we use the remainder to support several of the best students full-time for 10-15 weeks during the summer. They are treated basically the same as work-study students during that time, although no work-study funds are used in their support. They can work a maximum of 40 hours per week, and they retain student (not employee) status. In this instance, they are paid bi-weekly and the income is considered taxable. We pay them on a sliding scale, from $7.00 to $10.00 per hour, depending on the length of time they have been with the program, their skill level, and the perceived importance of their project to the university. In this way, the students can earn between $2,800 and $6,000 per summer, which is the range of what most CMU students of that caliber would earn at a regular summer job in business or industry.

---

[1] Quick calculation will show the reader that all the projects described in Section 3 were made possible with a total budget to date of $100,000.

6.1. Financial Aspects - for the students

In funding the program, the only restriction imposed on it by the university administration was that all funds had to be passed on directly to CMU undergraduates as financial awards. We were also asked to do our best to ensure that students would receive the maximum financial benefit from the awards: if possible, they should be shielded from having to pay taxes on the awards and from having any need-based financial aid reduced as a result of receiving the awards.

We worked with the Financial Aid office to develop an appropriate set of procedures for the program and to ensure that as many undergraduates as possible were eligible for consideration. In many instances, our guidelines are more open than most. For example, not only are age, sex, religious affiliation and ethnic background not a factor in determining eligibility for CSPEC, neither is citizenship. (Many other awards are available only to citizens of the United States.) Although we agreed to give strong preference to full-time students (who compose over 98% of CMU's undergraduate population), there are circumstances under which we will give an award to a part-time student. Award recipients remain classified as students; they do not receive any "employee benefits" as a result of participation in the program (and the program, as a result, need reserve none of its funds for benefits; it is able to use all the monies for the scholarships themselves). We have been able to meet the legal requirements so that awards during the semester are tax free. Summer awards, however, being based on 40-hour work weeks, are considered taxable income under the law.

The scholarship awards are based solely on "merit," never on "need." However, if a student does receive financial aid from the university (2/3 of CMU undergraduates do), there is no way we can completely shield the student's aid package from the effects of receiving an award. However, we have made special arrangements with the Financial Aid office that allow each student to negotiate individually the specific impact it will have. Thus, one student may elect to have the award affect the amount of loans s/he is expected to take out, thus reducing the total indebtedness incurred while an undergraduate yet keeping intact the amount of work-study money available. Another student may feel that, with coursework and the scholarship projects, s/he has fully committed the time available over the semester, and will therefore choose to have the scholarship reduce the amount of work-study expected while leaving the loan portion intact. Yet another student may choose to spread the effect equally across the different parts of the financial aid package. And still another student will have adequate personal and family resources so that no financial aid package is available; that student's financial aid status will be unaffected by the scholarship.

We have begun to insist that all participants start out with a "half" award for their first semester in the program. If they complete their service early in the term, and their schoolwork does not seem to be suffering from the additional responsibilities of the project, we will authorize an additional fractional award later in the semester. If, for example, they remain on the Dean's List and insist that they can handle a full 200 hours' worth of service, we will likely give them that for subsequent semesters. If, however, they have only average grades, we will continue to authorize a series of fractional awards while monitoring their academic progress. And if they seem to have, or to develop, academic difficulties, we ask them to give top priority to getting their coursework in order. Only when that appears to be under control[1] do we have them begin, or continue, with work on their educational computing projects.

Each semester, we review the students' work and discuss possible plans for continuation or expansion of their projects or for changing to a new project. Once they are accepted into CSPEC, students may have a reasonable expectation of continuing offers of support in subsequent semesters.

6.2. Financial Aspects - for the sponsors

Two other issues of the financial arrangements are more relevant to the department that administers it than to the students themselves. First of all, the scholarships go as direct awards to the students, and no institutional "overhead" is charged on this particular account. That's the good news.

The other side of using all the funds for student awards is that none of the funds can be used to support the salary of the program's director nor any of the project sponsors. The program's director is a full-time member of the research staff at CDEC. She spends about half of her time administering the scholarship program, teaching the associated courses, directing selected students in independent study projects, recruiting and screening potential program participants, and supervising those scholars working on CDEC-sponsored projects. This is considered to be a major part of CDEC's service to the university, so the associated portion of her salary comes from the department's general operating budget. For faculty or staff members from other departments within the university who also agree to sponsor students within the program, we generally hope it is the case that having the students available will lighten the burden on the sponsor, rather than increase it. In many cases, the students are providing a means for the sponsors to gain some exposure into new ideas or techniques, where it would take them much more time to explore such areas on their own. In fact, it appears that sponsors' willingness to take on CSPEC students increases with experience.

---

[1] We did have one student who became so involved with his project that he took a leave of absence from his studies to work full time. He considers this a sign of a highly successful student project. We think that definition is debatable, and continue to encourage him to complete his undergraduate program, even while recognizing that his current employer does hold his work in high regard.

7. Ideas we have revised

As stated earlier, our initial plan was that we would have about twenty students on full awards each semester. We have found that a better design is to have about a dozen at any one time during the school year, on full or fractional awards, and about half a dozen full time during the summer.

- Since the students all basically work on different projects, it just takes up too much of one person's time to supervise many of them adequately and consistently during the school year.
- While some students do just fine with the responsibilities of a full courseload, not all do. Fractional awards allow each student to participate according to the best of his or her
- Even when everything is going wonderfully for students, adding yet another project to their schedules during the academic year can occasionally disrupt the balance enough that each part suffers a measurable amount. Allowing some of the students to do the major portion of their work during the summer, when they can devote their full concentration to the project at hand without having to worry about the pressures of schoolwork, often increases their productivity tremendously.

We initially thought that the courses would be a necessary part of the program, introducing each new scholar (and other, potential, scholars) into the "culture" of educational computing. While it is true that such activities did prove invaluable when the program started, they have not continued to be as important over time.

- The success of the program to date does owe a lot to the first round of courses.
- Although we do not regularly offer them in large group format, it is still possible, for those students who want or need to become assimilated into this new realm, to sign up with the CSPEC director for a directed study (for credit, not money) either before or during their time in the scholars' program.
- The university is currently reorganizing many of its computer-related courses (including programming, problem solving, instructional design, software engineering, etc.). It is hoped that the new plan will allow us to continue to offer the students a variety of good courses, while also providing the departments with a course numbering system that provides a mechanism for recognizing the contribution of such courses toward graduation requirements. Our courses will be offered on an occasional basis through this program.
- Even though we have reduced the significance of the course offerings in the overall design of the program, we do remain open to revising or reinstating them the moment we ascertain that the benefits would at least balance the costs of such action.

We initially thought that all students would generate their own projects.

- Some students do, and do quite well.
- Other students have appropriate backgrounds and interests, but just do not have enough overall experience to be able to generate innovative yet manageable projects. By teaming them up with existing projects in need of a helping hand, we are able both to supplement our departmental activities in the realm of service to the university and to provide the students with a vehicle for exercising and expanding their skills.
- We doubt that we would have been able to adequately supervise all the students who have been involved in the program if they had all been working on projects they generated themselves. Having some of them sponsored through other departments has helped to increase the program's success.

We continue to explore the best mix of individual and group projects.

- As long as there is an appropriate "chemistry" between the people involved, a team consisting of one or more faculty or staff members plus a student scholar will usually work.
- Teams composed solely of students are often fine for small, short-term projects.
- On larger and longer projects, the students often lack the experience necessary to work collaboratively (that's considered "cheating" if they do it in most of their classes!), so either the project gets bogged down or the sponsor spends an inordinate amount of time sorting out the relevant difficulties. We continue to explore ways to solve this problem.

There is one potential problem area of which we are continually aware, although we have been lucky to avoid any major difficulties so far: maintaining students' projects once they have graduated and left the university.

- One project, Prisoners' Dilemma, was so successful that it is now being sponsored by full-time staff members within the department where it was started.
- Several smaller projects have served as good training vehicles for new scholars; the number of such projects so far, however, has remained small enough that this has been one workable solution. We do not expect the ratio of the number of projects needing maintenance to the number of new scholars needing training to remain small forever.

This issue interacts with the earlier one on courses. We have considered adding a course on software engineering, but have not made a final decision on that. Particularly with no School of Education at CMU, our courses on instructional design fill an important need on campus; we have been able to make them both rigorous and within the reach of students other than just those in a science or engineering department's computer science track. With software engineering, however, some relevant courses are already available, although they are geared more towards the graduate or professional student with a high level of technical sophistication. Since CSPEC is oriented toward expanding opportunities in educational computing for students in all disciplines, we are still struggling to select appropriate topics and to find ways of effectively teaching them to relatively naive software developers. Please let us know if you have any ideas or suggestions on this issue!

8. Conclusion

The CDEC Scholarship Program in Educational Computing is consistent with CMU's ideal of liberal / professional education. It serves as a paradigm of student involvement as beneficiaries of and resources for all of CDEC's major agendas: research and development, technology transfer, and teaching and educational service. Having just celebrated its second anniversary, we already have ample evidence of the power of this type of collaboration with our students. We are proud to be able to salute their successes so far, to support their continuing growth and development, and to encourage them to help us find ways to further improve the program.

9. Acknowledgements

This paper expands on the information presented (verbally only) on two different occasions in October, 1987. The first was a session, *Students as Developers of Educational Computing Projects*, at the Consortium for Computing in Undergraduate Education (C-CUE) conference held in Pittsburgh, PA, on October 17-19, 1987. The second was a seminar in the CMU Academic Services series, *CSPEC: Where students get paid to learn and have fun!*, on October 23, 1987. The author wishes to thank the CDEC Scholars who helped in preparing and presenting those talks, including **Linda Branagan** (Junior, Professional Writing; joined CSPEC in the Spring of 1987), **Leonard Dickens** (Senior, Applied Mathematics / Computer Science; joined CSPEC in the Fall of 1987), **Wayne Mesard** (Senior, Psychology; served on the task force that developed CSPEC in 1985 and then joined the program as a Scholar in the Summer of 1987), and **David Thompson** (Senior, Chemistry / Computer Science; with CSPEC since its beginning in the Spring of 1986). The author also wishes to thank **Preston K. Covey** (Director, CDEC, and CMU Vice Provost for University Studies) and **William Y. Arms** (CMU Vice President for Academic Services) for their continued support of the Scholars' Program.

10. About the Author

Carol Scheftic has a B.S. in Mathematics from Carnegie Mellon University (1971) and an M.A. in Mathematics Education (1973) and Ph.D. in Educational Communications & Technology (1985) from the University of Pittsburgh. She has teaching experience at all levels from primary grades through postgraduate professional education. Her main research interest is in students' learning of problem solving strategies. She returned to CMU in September, 1985, joining the Center for Design of Educational Computing as a Research Scientist to direct a project on *Powerful Computing for Education in Mathematics and Science at the Basic Professional Level*. In October of 1985, she became a member of the task force that developed the design for the CDEC Scholarship Program in Educational Computing. She has been director of CSPEC since its beginning in December, 1985.

11. References

Morris, J. H., Satyanarayanan, M., Conner, M. H., Howard, J. H., Rosenthal, D. S. H., & Smith, F. D. (1986). Andrew: A distributed personal computing environment. *Communications of the ACM*, Vol. 29, No. 3, p. 184-201.

Scheftic, C., & Branagan, L. G. (1987). *The CDEC Scholarship Program in Educational Computing* (Technical Report). Pittsburgh, PA: Carnegie Mellon University, Center for Design of Educational Computing.

Sherwood, B. A., & Sherwood, J. N. (1986). CMU Tutor: An integrated programming environment for advanced-function workstations. *Proceedings of the IBM Academic Information Systems University AEP Conference* (San Diego, CA, Section 4, p. 29-37). Milford, CT: IBM Corporation.

Trowbridge, D. (1986). A sampler of educational software at CMU. *Proceedings of the National Educational Computer Conference (NECC '86)*, p. 135-142.

Trowbridge, D., Larkin, J., & Scheftic, C. (1987). Computer-based tutor on graphing equations. *Proceedings of the National Educational Computer Conference (NECC '87)*, p. 28-33.

12. Appendix: Dimensions of the Scholarship Program

Time Period	Total Number of Participants	Number Repeating	Number New
1985-86			
Spring (only)	23	n.a.	23
Summer	9	8	1
1986-87			
Fall & Spring	17	13	4
Summer	8	5	3
1987-88			
Fall (only)	11	6	5

Table 1: Number of Students Participating in CSPEC

Time Period	student generated project	----- sponsored by -----		
		CDEC	CDEC & Faculty/Staff	Faculty/Staff
1985-86				
Spring (only)	--- coursework and preliminary projects ---			
Summer	2	3	1	3
1986-87				
Fall & Spring	1	6	4	6
Summer	1	2	3	2
1987-88				
Fall (only)	2	3	2	4

Table 2: Sponsorship of CSPEC Projects

RESEARCH PROJECTS

David W. Brown
Vanessa Evans Huse
Jho-Ju Tu
Diane McGrath

ABSTRACT: Improvement of the Knowledge Base Through Use of a Graphical Database

David W. Brown, Doctoral Student, Department of Curriculum and Instruction, College of Education, Kansas State University

The purpose of the research is to examine the effectiveness of a graphics database on the knowledge retention, understanding, and attitudes of secondary/post-secondary students. Specifically in the fields of computer science and science, it is projected that correct mental models may be enhanced via the use of such a database in the presentation of abstract material. The software package utilized will have not only the standard database procedures, but will allow the inclusion of pictures, line drawings, icons, and text. Relational capabilities will allow conceptual linkage of the material to be presented to the student.

The Apple Macintosh system and the Business Filevision database is utilized for this research. This combination of hardware/software is chosen due to its user-friendliness, agreeable human interface (i.e., use of the mouse, pull-down menus, etc.), and flexibility in the area of graphical representations. Subject areas covered are areas of computer science, physical, and/or the biological sciences.

The students prior to the treatment, are given an introduction to the use of the hardware components and the operation of basic database functions such as display, search, and sort. All groups make use of the computer with the control group being given only textual representation of the concepts.

Following each treatment, a pencil and paper test is administered to determine knowledge retention and conceptual understanding. A second short survey, along with qualitative observation, is used to assess the student's attitudes toward this form of computer learning.

ABSTRACT: Implications of Right/Left Brain Research on Computer Education

Vanessa Evans Huse, Ed.D., Lon Morris College, Jacksonville, TX

Purpose of the Study: The primary purpose of this study was to determine whether a significant relationship existed between success in a college introductory computer science course and several intellectual and non-intellectual factors. Intellectual predictor variables which were considered included previous grade point average, composite ACT score, verbal ACT score, and mathematical ACT score. Non-intellectual predictor variables considered were level of fluency, flexibility, originality, elaboration, and primary brain dominance in regard to the mode of information processing.

Procedure: A total of 105 students were selected from two state universities in Texas: East Texas State University and Sam Houston State University. The Torrance Test of Creative Thinking and Brainscan were administered at the beginning of the spring semester of 1986. All grades and scores were obtained by the researcher from the registrar of each institution. At the end of the semester, final grades were reported by the instructors.

The statistical procedure analyzed the relationship between the criterion variables, final grade in the first semester computer science course, and the set of independent predictor variables. Multiple regression was used to determine if particular relationships existed between intellectual and non-intellectual independent variables and the dependent variable the final grade in the course. Beta coefficients were calculated so the the value of the coefficient for each predictor variable showed the function of the correlation between predictor variables used when necessary to narrow their choices of predictors.

Findings: The data were analyzed to the purpose and the research questions in the study. The first research question dealt with intellectual factors which might be used to predict success of students in an introductory computer science course. Of the variables used as predictors, only grade point average could be used as a valid predictor for success using multiple linear regression with significance set at the 0.05 level. Other intellectual factors which were considered but which showed no relationship included total ACT, mathematical ACT, and verbal ACT scores. However, mathematical ACT scores had a significance of 0.0595.

The second research question in the study dealt with several non-intellectual factors which might be used to predict success of students in an introductory computer science course. This included digital and analog dominance and levels of fluency, flexibility, originality and elaboration. None of these variables showed significance at the 0.05 level; however, the elaboration score on the Torrance Test had a significance of 0.074 while the fluency score had a significance of 0.1121.

Conclusions: Based on the results of analysis in this study, the following conclusions are presented:

1. With significance set at 0.05, the only intellectual factor which can be used to predict success of students in an introductory computer science course was grade point average. This conclusion agrees with many other researchers, including Bloom and Peters (1961), Beavers (1968), and Konvalina (1982).

2. Although the significance level for mathematical ACT scores did not reach the 0.05 level, the significance level showed that the mathematical ACT score may have some bearing on success in an introductory computer science course. This agrees with other researchers including Peterson and Howe (1979) and Konvalina (1983).

3. With significance set at 0.05, none of the non-intellectual factors which were examined can be used to predict success of students in an introductory computer science course. These included digital and analog dominance; and levels of fluency, flexibility, originality, and elaboration.

4. Although the significance level for elaboration and fluency did not reach the 0.05 level, some relationship was indicated. Both of these characteristics have been isolated as factors which programmers must have to be successful by Canning (1957) and Upshall and Reland (1958).

These conclusions agree with many other researchers including Bloom and Peters (1961), who believed that past achievement is often the best index of future achievement. Others such as Beaver (1968), Correnti (1970), Peterson and Howe (1979), and Wechslerr, Stephens, and Konvalina (1982) found that grade point average was the best predictor for success in an introductory computing class.

ABSTRACT: The Relationship Between Learning Programming and Problem-Solving Ability

Jho-Ju Tu, Graduate Assistant, Dept. of Secondary Education, College of Education, University of Illinois, Champaign, IL 61820

How to increase problem-solving ability has been considered an important subject by educators for several decades. Many claims from the psychology of programming have provided strong support for the utilization of computers in teaching problem-solving. Recent research findings regarding the increase of problem-solving skills for students learning computer programming are, however, conflicting. The question of whether the skills students learn from a programming course will improve students' problem-solving abilities remains open.

The students in three introductory programming courses (CS103-*FORTRAN*, CS105-*PASCAL*, CS106-*BASIC*) offered by the Department of Computer Science at the University of Illinois Urbana-Champaign in the fall semester of 1987 served as the subjects for this one-semester long study. They were given pretests and posttests on problems in seven categories: word problem translation, word problem solution, following procedures, following directions, logical reasoning, visual ability, and verbal ability. This project aimed to clarify the relationship between programming experience and problem-solving abilities. To be presented will be new findings on (a) whether a student's problem-solving abilities can be improved while learning to program, and (b) whether different programming languages such as *BASIC*, *Pascal*, or *FORTRAN* have different effects on student's problem-solving abilities.

ABSTRACT: Hypertext as a Medium of Instruction for Complex Material

Dr. Diane McGrath, 261 Bluemont Hall, Kansas State University, Manhattan, KS

It has been shown that students learning complex, ill-structured subject matter often develop misconceptions about that subject matter based on the use of simple analogies to teach beginners (Spiro, et al., in press). These misconceptions persist long after instruction has introduced and explained the complexities and irregularities of the subject. A number of authors (Wittgenstein, 1945; Minsky, 1985; Spiro, 1987) have written of the need to deal with complex material in some nonlinear fashion in situations in which the ideas are web-like, involving many interconnections. The development of hypertext has provided us with a medium for transversing complex material from several different directions. If this type of instruction can avoid the pitfalls of persisting misconceptions and thereby help students to better understand complex material, it will prove to be of great significance to educators because hypertext lessons can be written by faculty who have no programming experience, and so the solution to this problem can be widely available.

The research to be presented will show the results of a comparison of the acquisition and retention of complex material by college-age learners under three conditions: (a) presented as straightforward text and pictures, as in a traditional book; (b) presented as straightforward text and pictures, as in a traditional book, but on the computer screen, and (c) presented as a hypertext document. It is anticipated that students who are able to study the material in the form of hypertext will show the best understanding and the fewest misconceptions about the subject and that students learning the materials in a straightforward, linear, book-like fashion, whether using printed material or viewing it on the screen, will show the least understanding and most misconceptions.

POWER TOOLS FOR ALGEBRA PROBLEM SOLVING

Nancy Roberts
Richard Carter
Frank Davis
Lesley College

Wallace Feurzeig
Bolt Beranek and Newman

ABSTRACT

The Logo-algebra project attempted to test how much algebra could be taught to sixth graders given the creation of courseware using advanced technology suitable for the sixth grade classroom. The project created four curriculum modules covering the algebra topics of variable, equation, function, and graphing. The Logo language was intertwined into two of these modules to see if learning and using the language in an algebra environment would support the learning of algebra. Two extensive software environments for the Macintosh computer were created to support these four modules, the Marble Bag Icon Laboratory and the Algebra Workbench. In additional, several smaller pieces of software were developed for teaching more specific skills such as reading and creating graphs. Three modules were used extensively with the students, while the fourth only briefly and therefore not evaluated.

Both a quantitative and qualitative evaluation were conducted. These evaluations suggested that the sixth grade students could learn algebra concepts in the context of the microworlds, but did not demonstrate a transfer of these concepts to a more traditional algebra format. After seeing the pedagogical power of the Marble Bag microworld during the first cycle of pilot testing, a Weather Station microworld was created to aid in teaching some graphing skills that proved difficult during this first cycle. The Weather Station microworld showed again how effective such learning environments are.

The evaluation also suggested that it was too difficult for sixth graders to learn both the Logo language and algebra at the same time. When Logo was simplified through the creation of super-procedures, programming became a useful tool for the learning of algebra.

INTRODUCTION

Since calculators and computers are approaching reasonable numbers in our schools, now seems to be the opportune time to re-evaluate the content of our mathematics curriculum. The first six years of school mathematics has been traditionally spent on learning number facts and the algorithms for adding, subtracting, multiplying, and dividing. If these tasks can be considerably shortened by having students use machines, what mathematics should then be taught in these early grades? Can higher level mathematics, such as algebra, be moved to the lower school with the aid of computers? Can the introduction of programming, and especially Logo, support more advanced mathematical thinking at an earlier age? These are the questions this project has begun to investigate.

Specifically, we developed curriculum supported by software to introduce algebra to sixth graders. Two classes of sixth graders studied algebra using our materials, each for one half a year. This curriculum included four modules: Unknowns and Equations, From English to Algebra, Functions and Graphing Equations, and Using Algebra. Time did not permit extensive teaching of the fourth module. It was therefore not part of the evaluation.

Several new software environments support these modules. These include two major pieces, called the Marble Bag Icon Laboratory and the Algebra Workbench. In addition there are several smaller programs including pieces similar to the commercial software Globs, Guess My Rule, and Expression Writer in addition to our own One-liners and the Weather Station. The Logo language is used extensively as an additional computer environment in this curriculum.

Three fundamental algebra ideas were investigated in this project:

1. Students more readily develop understandings of complex ideas if they are helped to make their thinking explicit. Programming involves students in articulating what they know, externalizing the process of solving (or generating) a problem, teaching the machine, and, in some activities, anticipating how to teach other students to solve problems. Throughout the modules students write programs in Logo to generate algebraic equations or model functions.

2. New abstractions, such as those involved in learning algebra, should be tied to well-understood concrete events or to meaningful metaphors. One strategy for teaching abstract ideas is to have students experience multiple representations of these ideas and learn to translate between the different representations. In this curriculum students translate among several representations of an algebraic expression including English, pictorial representation, Logo programming code, and conventional algebraic notation. To facilitate the use of pictorial representations, icon-based environments on the Macintosh are used.

3. Three important underlying algebra concepts are within the cognitive ability of sixth graders when operating in the computer microworlds: the concept

of variable, the composition of functions and the reversibility of operations.

CLASSROOM IMPLEMENTATION

The pilot study was conducted at the Hanscom Air Force Base Middle School. The algebra course was taught to two classes of sixth graders, including all the students in that grade who had a free study period. The curriculum encompassed one semester. The students who did not have a free study acted as the control group.

Module 1 - Unknowns and Equations

Module 1 begins with a simple "guess my number" game (based on work originated by Max Beberman). A marble bag is the metaphor used to represent the algebraic unknown rather than the abstractions of *X*s and *Y*s. The module begins with the concrete objects of bags and marbles. Students are divided into groups, with one person in each group designated as *secret keeper* and asked to put a secret number of marbles in the bag. A story is then told beginning with "Think of a number," which is the English equivalent of putting some number of marbles in the bag. The story might then continue:

Add three;

Multiply by three;

Subtract your original number.

The class is then shown how to represent stories on the blackboard using pictures of bags and marbles as shown in figure 1.

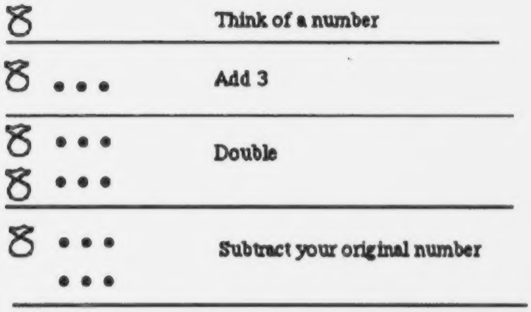
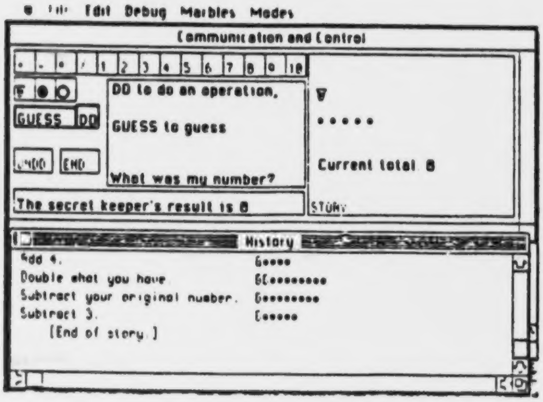


Figure 1. Blackboard Recording of Story

The students discover together the "trick" of finding out the unknown number of marbles in the bag by *undoing* the story. To practice creating and solving marble bag stories without needing a group of students with whom to work and without having to write out the representations "long hand", the Marble Bag Icon Laboratory is introduced. This laboratory allows students to build marble bag stories and manipulate the results, using Macintosh computers. See figure 2.



DO to do an operation,

GUESS to guess

What was my number?

The secret keeper's result is 8

Current total 8

Add 4.

Double what you have.

Subtract your original number.

Subtract 3.

[End of story.]

Figure 2. Marble Bag Icon Laboratory

The next step is to introduce the second iconic metaphor, the balance. The teacher starts by manipulating a real balance with the class and then moves to the software. The first computer laboratory just allows students to manipulate bags and marbles on the balance so they can become familiar with the concept of balance and the software. The software then generates problems for the students and challenges them to find the number of marbles in one bag (to solve the bag problem) in the fewest possible steps. This can always be done in two steps if the students do the same operation to both sides of the equation at each step by first subtracting the marbles and then dividing the bags.

At this time standard algebraic notation is introduced by substituting numbers for multiple bags and marbles and showing how a bag is much more easily drawn without its top and bottom, producing a character resembling an X. The students then practice telling and recording marble bag stories using standard notation or whatever combination of notation they prefer. Students used combinations of all three representations: English, icons, and standard algebra as shown in figure 3.

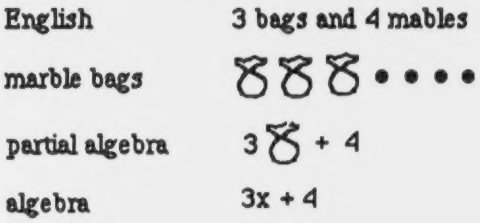


Figure 3. Examples from Student Work

Using the Marble Bag Icon Laboratory, stories can be represented in English, with marble bag icons, and with standard algebraic notation. Students are asked to go back and forth between these representations of their stories. For instance, *"Think of a number "* is called *X* , *triple what you have* results in *3X*, *add 5* is *3X+5*, and so forth.

During this module simplifying expressions by combining terms is taught followed by using parenthesis.

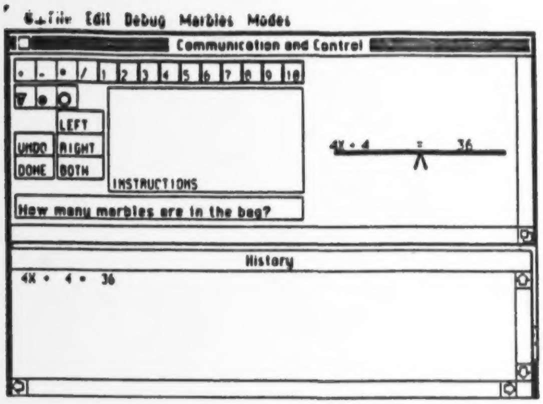
The metaphor for parenthesis (factoring) is packaging and removing parenthesis (distribution) is unpackaging. The students use shoe box covers as game boards to physically package and unpackage bags and marbles. Practice in packaging, or using parenthesis, is provided by the software program, *One-liner*. Here the computer generates a Marble Bag type story and the student's task is to write the story, in formal algebra notation, as one line.

The next step is to introduce students to solving equations that contain variables (MarbleBags) on both sides of the balance. The Balance Laboratory, which can show the equation both as icons or as standard algebra on the balance, is again used. See figure 4.

A second software environment, the Algebra Workbench is introduced here in its calculator mode. The students enter their equation into the workbench and solve it, step-by-step, using the software in a calculator fashion. The Algebra Workbench is an intelligent tutor, but the class did not have time to use much of the tutoring capabilities. A description of the Workbench follows the description of the classroom tested modules.

Module 2 - From English to Algebra

The second module begins with students writing Logo procedures to manipulate language, in particular procedures to generate gossip. Manipulating language is a non-threatening and engaging entry into programming. Through generating gossip the machine imagery that later serves as a model for functions is introduced.

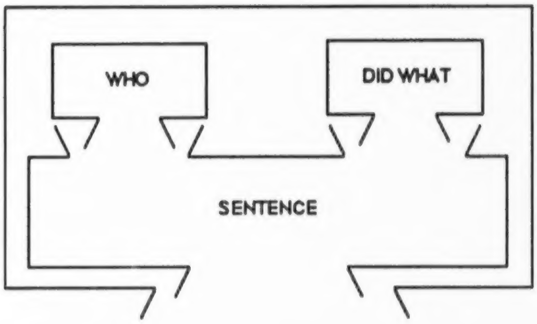


How many marbles are in the bag?

4X + 4 = 36

Figure 4. Balance Laboratory Displaying Standard Algebra

In the first part of the module students generate examples of gossip on the blackboard and work together to look for a pattern. Once a pattern has been found they learn how to represent it in terms of machine imagery and then how to directly translate the pattern into a Logo program. See figure 5.



```
TO GOSSIP
  OUTPUT (SENTENCE WHO DID.WHAT)
END

TO WHO
  OUTPUT PICK [CHRIS RICKY RONALD
   NANCY]
END

TO DID.WHAT
  OUTPUT PICK [[ EATS ICE
  CREAM][FLUNKED MATH][IS DIVORCED]]
END
```

Figure 5. A Gossip Machine and Logo Program

The early focus on language leads to creating gossip programs with variables and interactive quiz programs, beginning with simple arithmetic quizzes, and moving onto algebra quizzes. The machine imagery is used again to generate quizzes as it is used to generate gossip. See figure 6.

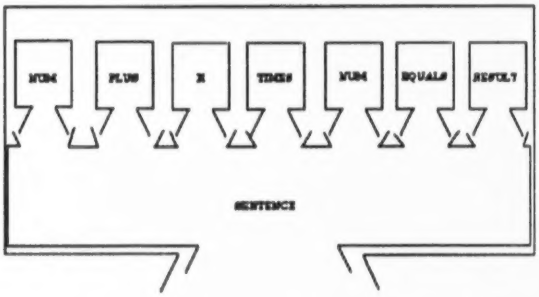


Figure 6. Machine to Generate an Arithmetic Quiz

In addition to having a structure that supports algebra—variables, symbolic manipulation and equations—the quizzes afford students the opportunity to change roles from recipient to teacher and developer. The challenge here is for students to solve the quizzes their own Logo program generates as well as the quizzes generated by other classmates' programs. Here the Algebra Workbench is used as a tool for algebra equation solving.

Module 3 - Functions

Module 3 is an introduction to functions and graphing equations. In this module students are engaged in mathematics as a tool to solve "real" problems in simulated environments. The first simulation involves students as cryptographers. Students write Logo procedures to encode and decode words and sentences, beginning with simple codes. For instance, one procedure might produce:

MPHP BMHFCSB

which is the coding of:

LOGO ALGEBRA.

The code is generated by offsetting each letter by one in the alphabet, *a* becomes *b* becomes *c* . The utility of using numbers to represent the place of letters in the alphabet is developed and students write Logo functions for encoding and decoding messages. Students also make up their own code rules and write Logo functions, both for coding and decoding. Undoing becomes one way to think about decoding. The codes also can be represented and constructed using the machine metaphor.

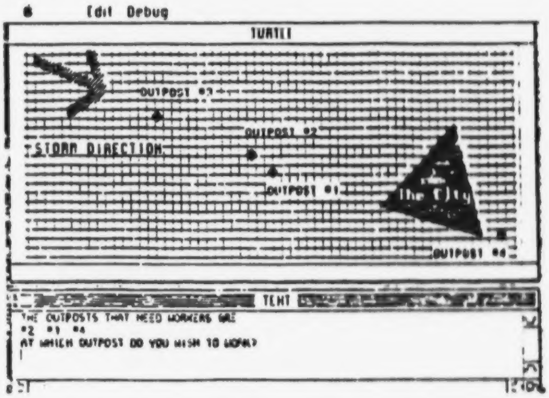
After working with codes based on addition and subtraction, codes based on multiplication are introduced. Encoding proceeds by multiplying a letter's alphabetic number by some factor—perhaps 2. In this case H (being letter number 8) becomes P (letter number 16). A is used as the letter after Z to handle doubling that goes over 26. The codes create function-like relationships such as $Y=X+b$ and $Y=aX$. Combining multiplication and addition in a code can be represented by an equation in the form of $Y=aX+b$. Finding the coding rule, given a coded and decoded message, is a challenge addressed by making graphs and doing a mathematical analysis of the results.

Another part of Module 3 uses the *GUESS MY RULE* function game. This begins with the teacher creating charts of inputs and outputs and students having to guess what the "rule" (function) is producing the output. This moves to Logo-based games where a program generates outputs based on students' test inputs. The students then guess the rule and write Logo procedures to reproduce the input and output patterns. Students also are asked to write their own function machine procedures to challenge other students. A graphing utility is part of this activity so that students can see a graph of any input/output chart to aid in figuring out a rule.

The second simulated environment puts students in the role of weather predictors. The Storms environment shares similar goals and themes with the codes environment—an introduction to functions and graphing. The software created centers around a tool environment where students become data analysts for a weather center. The computer shows lightning followed by the sound of thunder. The weather center person can calculate the distance of the storm from the city based on the fact that the number of seconds between the lightning and thunder divided by five is the distance away of the storm, or $y=x/5$. In addition, the number of miles the weather center is from the city needs to be added or subtracted. See figure 8. The students are next introduced to a way of making their work easier by accumulating their storm data in a chart. See figure 7. Next, as in the codes simulation, the data points are graphed and the students now have a new tool to calculate storm distances.

To give students experience with different slopes, the students are assigned to the interplanetary weather research team working with weather on distance planets where the lightning/thunder time rules are quite different; for example, instead of dividing by 5 you multiply by 3. A new software tool is then provided that allows students to plot points or equations, turn plotted.

points into a line, use the mouse to determine the rule of the line, and then read data from the graph. Students may then compare the graphs of different rules such as, How does changing the lightning rule change the graph?



```
THE OUTPOSTS THAT NEED WORKERS ARE
#2  #1  #4
AT WHICH OUTPOST DO YOU WISH TO WORK?
```

Figure 8. Screen from Storm Center

Module 4 - Using Algebra

The fourth module focuses on problems whose solutions are aided by the application of linear algebra. Students write Logo template programs to generate random story problems for a given equation. The Logo function machine metaphor is used to help generate and solve word problems. Using the machine icons students build simple function machines and connect these machines together to create more complex machines, modeling the composition of functions. An example is the following punch party problem.

> The sixth grade class wants to give a party. They want to know how many quarts of lemonade to make. The paper cups they already have hold 6 ounces each. They have invited 48 people to come.

When asked, "What do you need to know to solve this problem?" one student responded that you need to know how many ounces in a quart. Another said, you need to know how many glasses in a quart. A third student said, you need to figure out how many ounces will be drunk. The idea is to use Logo and machine icons to create machines that solve various parts of the problem and then put the pieces together to answer the original question. In this example, a machine to translate ounces to quarts might be the first step. Then a machine to translate people to number of ounces can be built. Each of these machines can then put together to create a "*people to quarts* " machine as shown in figure 14.

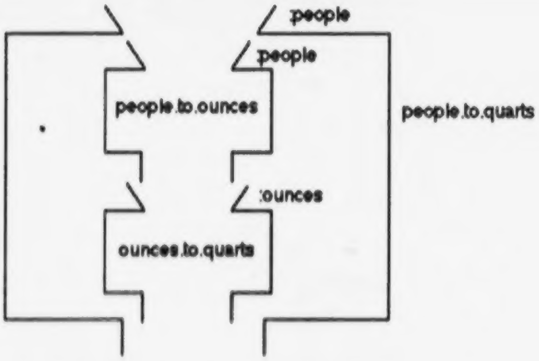


Figure 14. Punch Party Machine Representation

The machines can be turned into the Logo program:

```
TO PEOPLE.TO.QUARTS   :PEOPLE
  OUTPUT  OZ.TO.QUARTS  PEOPLE.TO.OUNCES
      :PEOPLE
END
```

Once the machines are built they can be used to solve other similar problems. Given a set of guided experiences with several of these problems, students find it easier to take other similar problems apart. Having students approach problems by building simple machines helps them isolate different aspects of problems and aids them in parsing problems into useful pieces. Interpretation and extension of each problem results as the class tries to construct solutions. In the lemonade example someone might ask, "What if the people who come to the party drink more than one glass of lemonade?" (a June party in Atlanta). This leads to the question, "Can we fix our machine so it can handle this possibility? What do we have to change if each person drank two glasses?"

A further step in creating these machines is to ask students if they can build more general purpose machines, so they could handle different sized cups, or different numbers of cups per person in a new lemonade problem. By replacing constants with variables students create more general purpose tool programs. The machines students create become tools in a tool kit for exploring (and generating) related word problems. This opens the way for the creation of sets of word problem-solving tools centered around different domains. Because of time constraints, very little of this module was tried with the pilot students.

THE ALGEBRA WORKBENCH

The Algebra Workbench is an intelligent instructional tool designed to provide the student with a structured algebra exploration environment. The Workbench is an operational system in prototype stage. It has two parts, the calculator and the expert equation solver. The calculator can be used by the student to solve equations by automatically performing operations on equations or algebraic expressions. The student can perform operations (addition, subtraction, multiplication, or division) on both sides of an equation, or invoke functions (Do Math, Remove Parentheses, Collect Terms, Expand Terms) on parts of an equation. By providing the means to perform computations, the calculator allows students to focus on strategies for solving problems without needing to worry about the arithmetic. The calculator allows students to try more than one solution path through a problem, branching at any point along the way. See figure 9.

When the student is unable to figure out the next step, he or she can call upon the *advisor*. This element of the Workbench suggests one possible next step to the student. The advisor first states the goal, then how to reach the goal. The advisor does not evaluate the student's previous work, unless the student specifically requests the evaluation. Moreover, the advisor might not suggest the best possible next step, just a workable next step.

If the student encounters an entirely new problem that he or she has no sense of how to approach, the student, may call upon the *expert* component of the Workbench. The software now takes the student through the solution of the entire problem, first presenting the goal and then the solution for each step of the problem. Again, the software may not necessarily present the student with the most elegant solution path through the problem. However, the solution is always a correct solution.

Several issues have surfaced during our work with this piece of software. Such questions as, Should the software *know* the best solution through a problem at any point along the way, Should the software allow students to make mistakes, How small should the steps be when the computer presents a solution (when multiplying both sides of an equation to eliminate fractions should the cancellation be explicit) have been continually asked by the project staff. Many more additions to the workbench have been suggested by both staff and cooperating teachers. These include things such as:

- provide toggle switches so the teacher can turn on or off the advisor and expert;
- integrate graphing capabilities with the Workbench;
- provide an ability to use icons with the Workbench, such as bags and marbles;
- provide several algorithms through a problem so students may see more than one solution.

We hope to attend to several of these questions as we continue with this work.

EVALUATION

The first year of the project was devoted to software and curriculum design. During the second year these products were tested with the sixth grade students at the Hanscom Air Force Base School in Bedford, Massachusetts. A half-year course was developed and taught to two classes within the school. Both formative and summative evaluation was carried out during the second year. The formative evaluation during the first half of the year led to some curriculum changes in the second half. The intent of the evaluation process was to assess both the cognitive and affective learning of the children, and how this learning could be improved. See Appendix A for the complete evaluation report.

Three classes of students made up the sixth grade population and therefore could be configured as shown in Table 1 for the evaluation process.

Table 1

	Fall 1986	Spring 1987
Class A	1st Logo-algebra course	
Class B	Control Group	2nd Logo-algebra course
Class C		Control Group

A pretest was administered to the entire sixth grade in the fall. Unit tests were given at the end of each module during each half course. Post-tests were given at the end of each half course to the group being taught and the control group. The two post-tests had some different questions to reflect the curriculum changes made for the second half year. In addition to the formal tests, a project staff member was present in every class as an observer. Parents were interviewed at the end of the first half course and students were interviewed at the end of the second.

The pre- and post-tests attempted to evaluate the three major thrusts of the curriculum: to see if the students receiving the course could: understand algebra in its traditional form; learn algebra using the computer microworlds created by the software; and learn algebra by learning Logo. The tests looking at traditional algebra learning showed that children who took the course did significantly increase their ability to solve algebra equations, read graphs, and understand functional relationships. In addition, the May, 1987 post-test was also administered again to the students in the first half-year course, showing no significant change in their scores from their January post-test. This suggests that the students from the first half-year course had retained, at least until May, what they had learned previously. This evaluation of traditional algebra knowledge suggests that the curriculum had some success. During the first cycle of the project children moved from percent mean scores of 17% to 49% and during the second cycle from a percent mean score of 21% to 64%. In addition, there is an indication that children in the second course were able to perform better than those in the first cycle. This improved performance is partially accounted for by some modifications made in the second cycle.

The tests that looked at the children's ability to do algebra in the microworlds showed that students from both classes clearly could handle algebraic concepts in the context of the marble bag and balance microworlds. It is important to note the children also performed well on the part of that test that was written in traditional algebraic form. The tests that looked at the impact of learning Logo on acquiring algebraic concepts were mixed for the first class. These students clearly had trouble learning Logo, but not learning the algebra in its traditional form. As a result, modifications were made to the curriculum, de-emphasizing the Logo syntax by creating super-procedures for the children to use. The children in the second cycle performed well on all parts of the comparable test. It should be noted that the students did not use the computer for the Logo test, and were therefore in a different, and static, environment for only the test sessions.

For the third module, the curriculum was again revised based on the results of the first classes tests, which were quite mixed. The children could successfully understand coding and decoding rules, and their algebraic representation. However, when these same rules were represented graphically, children had difficulty translating to a correct algebra representation. Children had extreme difficulty in understanding and constructing Logo procedures that code and decode messages. Children, however, were able to handle problems involving the algebraic representation of a functional relationship with great success. For the revised module for the second class another microworld, a weather station, was created. Here the children used equations and then graphs to predict the distance away from a city of a storm. The rate of speed of sound and the position of the observer were varied to increase student challenges. The test for this module had three parts: Part I tested the students' ability to understand rates of change; Part II to analyze and construct graphs; and Part III to analyze and construct equations. The total percent mean score for this test was 80.4, suggesting that the children handled most of this test without difficulty.

A variety of statistical tests were made to try and determine if children were able to transfer what they had learned within the unique microworlds and programming tasks of this curriculum to the more traditional content of an algebra curriculum. A regression analysis suggests that they were unable to do this.

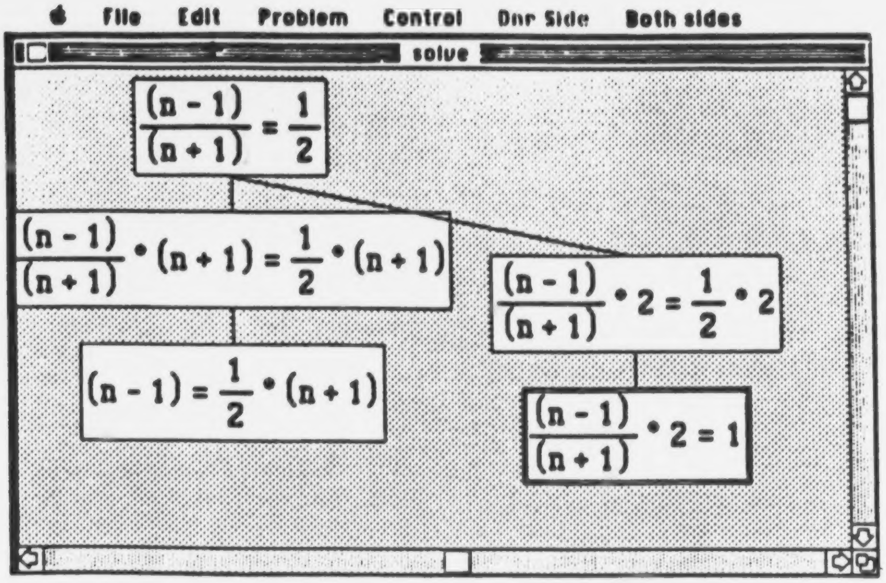
To summarize the evaluation from a quantitative perspective, there were many indications that children achieved the objectives of the curriculum—especially when these objectives were translated into the unique microworlds.

However, from the perspective of a traditional measurement of algebra skills the acquisition of curriculum objectives is not as evident.

In addition to the qualitative evaluation, there was an observer in the classroom during the entire year, taking continual notes of the activities. In general, these observations support many of the quantitative results. For example, the observations reveal the difficulty of moving from the content-dependent subject of the modules to the more general subject of algebra. In addition, these notes reveal a learning environment that depends upon independent discovery learning. This might explain some of the different patterns of performance found throughout the modules.

SUMMARY

The Logo-algebra project created two extensive software environments: the Marble Bag Icon Laboratory and the Algebra Workbench. The Marble Bag Laboratory was used with sixth graders for an entire year while the Workbench was used only briefly. Four curriculum modules were created to use along with this software and additional smaller pieces of software. Three modules were used extensively with the students, while the fourth only briefly and therefore not evaluated. Both a quantitative and qualitative evaluation was conducted. Both of these evaluations suggested that the sixth grade students could learn algebra concepts in the context of these microworlds. The evaluation also suggested that it was too difficult for sixth graders to learn both the Logo language and algebra at the same time. When the Logo language was simplified through the creation of super-procedures, programming became a useful tool for the learning of algebra. There was no evidence from the evaluation that the sixth graders were able to transfer the algebra they learned in the context of the microworlds to more traditional algebra.



$$\frac{(n-1)}{(n+1)} = \frac{1}{2}$$

$$\frac{(n-1)}{(n+1)} \cdot (n+1) = \frac{1}{2} \cdot (n+1)$$

$$(n-1) = \frac{1}{2} \cdot (n+1)$$

$$\frac{(n-1)}{(n+1)} \cdot 2 = \frac{1}{2} \cdot 2$$

$$\frac{(n-1)}{(n+1)} \cdot 2 = 1$$

Figure 9. Algebra Workbench Showing Branching

INTELLIGENT TOOLS FOR MATHEMATICAL INQUIRY

Richard Carter
Lesley College
The Compass (Education Division)
29 Everett St.
Cambridge, MA 02238
617-868-9600 ext 370

Wallace Feurzeig
John Richards
Bolt Beranek and Newman

Nancy Roberts
Lesley College

ABSTRACT

Intelligent tools for Mathematical Inquiry is a National Science Foundation funded project aimed at using the power of interactive computational environments to shift the emphasis in the teaching of mathematics from a body of facts and procedures to informed exploration and reflective inquiry. We believe the use of appropriate technologies coupled with new approaches to curriculum can foster the teaching of the skills of mathematical inquiry. We call our approach *inquiry by design*. In *inquiry by design* the student must be motivated to think actively about mathematics, to support this we plan to build tools that are designed to guide exploratory learning, tools that will support learners actively constructing their own knowledge. Currently we plan to work on three different types of intelligent instructional tools: an expert system/tutor, the *Algebra Workbench*, an intelligent instructional microworld, the *Marble Bags World*—and the design specifications for a visual programming tool, the *Function Machine Laboratory*. These tools will be implemented for use in teaching mathematical inquiry skills in the areas of algebra, functions, and mathematical modeling. This paper focuses on the rational for the project and the software tools we plan to develop.

INTRODUCTION: AN OVERVIEW OF INTELLIGENT TOOLS FOR MATHEMATICAL INQUIRY

The teaching of science is so much more than the imparting of knowledge, or imparting a certain method, but goes back to the really basic principles on which the Republic was founded: It is to try to give the student a vision of the beauty of truth...

I. I. Rabi

The development of a mathematically literate American population will come about only through helping young students experience the sense and value of real mathematics. The emphasis in schools must shift away from teaching mathematics as a body of facts and procedures. Instead, students must be introduced to the activity of mathematics as informed exploration and reflective inquiry. Mathematics should be presented actively through tasks and projects that are truly interesting and motivating. Interactive computational environments with powerful new capabilities derived from advanced work in computer science and artificial intelligence are ideally suited to the presentation of such tasks and the development of the desired knowledge and skills.

The use of appropriate computational environments can foster the teaching of the skills of mathematical inquiry. These are the true basic skills of mathematics, those on which the development of competence in computation and mathematical inference fundamentally depend. The teaching of these skills requires a community of intensively trained teachers who are comfortable and competent with the ideas and technology. It is time to develop a mathematical inquiry program strongly supported by intelligent technology and teaching, as a model of the kind of curriculum that is essential for transforming and revitalizing elementary mathematics education.

The purpose of this project is to develop such an approach. Much more than imparting a body of knowledge or a set of methods, our interest is in helping students to experience the sense and flavor of genuine mathematics and to develop skill in the activity and processes of mathematical inquiry. We propose to accomplish this partly through a laboratory presentation centered on mathematical modeling activities that span a rich variety of compelling problems and projects, and partly through the design and use of intelligent tools that support learning and empower students. The tools we plan are at the cutting edge of computer science and artificial intelligence. Advanced technology has been extensively applied to the development of prescriptive tutoring systems. We need to explore its use in creating structured non-directive learning environments so important for supporting the development of mathematical inquiry skills.

We therefore plan to develop:

1. Three distinct types of intelligent instructional tools—the *Algebra Workbench*, the *Marble Bags World*, and the *Function Machine Laboratory*—that exploit state-of-the-art methods to provide structured non-directive learning environments;

2. Three distinct mathematical inquiry modules, for Algebra, Functions, and Mathematical Modeling, to introduce mathematical inquiry into the curriculum; and

3. Intensive teacher training courses, to teach the teaching of the notions and art of mathematical

inquiry.

Inquiry by Design

Inquiry by design, has three distinct aspects. The first denotes the role of the instructor as designer: to motivate the student to think actively about mathematical issues. Polya's first principle of teaching is to make the student an active learner (Polya, 1962). It is necessary to design tasks and projects that stimulate students to ask questions, pose problems, and set goals.

The second aspect of *inquiry by design* denotes the role of the intelligent instructional tool as designer. The tools we are planning are designed to guide exploratory learning. Nondirective computer environments, called *microworlds*, have been created to enable students to probe and study the behavior of mathematical systems and to discover and extend their knowledge through such exploration. Microworlds can be used productively in this way by sophisticated students who already know how to learn. For others it is necessary to provide a structure and a set of plans to support student exploration. We have begun to provide such structure, through advanced techniques from artificial intelligence designed to aid the development of *informed* exploration and *reflective* inquiry without taking initiative or control away from the student.(Roberts, 1987) Students will not become active learners by accident, but by design, through the use of the plans that we structure to guide exploration and inquiry.

The third aspect of *inquiry by design* denotes the ultimate goal of the program; the student as designer. Learners must actively construct their own knowledge. Our approach focuses on the constructive aspect of design as an essential complement of analysis in a student's progressive construction of mathematical knowledge. Our experience is that the activity of designing and building mathematical objects and processes can be used to motivate students' understanding of mathematical structures and the sense and methods of mathematical inquiry. Programming languages with recursive functional constructs are the perfect instrument for supporting a constructive approach to developing mathematics. As is the case with microworlds, however, even the most accessible of such languages, Logo, requires a supporting structure to enable students to learn to use it easily and effectively. We will provide such a structure through the use of new iconic programming techniques designed to make Logo programming semantically and visually transparent.

Intelligent Instructional Tools and Mathematical Inquiry

The central focus of this project is the development of powerful new computer-based tools expressly designed to support the teaching and learning of mathematical inquiry skills throughout the elementary mathematics curriculum. Corresponding to three distinctly different ways of embedding intelligent support for open-ended inquiry activities, three major types of tools will be developed: an expert system/tutor, an intelligent microworld, and a visual programming language.

The first type of tool is based on the technology of expert systems and intelligent tutoring. This expert/tutor tool can be used to support well-defined problem-solving tasks in a way that allows students freedom to pursue their own solution paths, to request advice on subgoals and actions from the tutor, or to have the tutor demonstrate and explain the strategy and elaborated details of a solution. The tutor can critique the student's work and detect specific errors. It can solve problems posed by the student and explain its reasoning along the way.

A second type of tool is based on the technology of computer microworlds. Microworlds provide concrete manipulable representations of abstract mathematical concepts and processes. They often employ object-oriented methods to support iconic operations within dynamic graphic environments. They can be used as open-ended exploration environments with the goal of fostering an understanding of the mathematical structure and behavior of a system of objects and operations. Students can also use them to solve problems by creating and manipulating objects and operations. These activities can be supported by incorporating intelligent facilities for posing problems and critiquing solutions.

The most powerful type of computer tool is a universal programming language. Programming provides the most open-ended computational environment for mathematical expression, and can provide the richest variety of activities for the development of mathematical inquiry skills. In particular, programming can support modeling projects and sequences expressly designed to introduce the concepts and methods of mathematical inquiry. To realize this potential, however, it is necessary to develop tools to make programming skills accessible to students and teachers. We have designed the prototype of a visual programming environment that employs iconic constructs to represent the basic program structures of the functional programming language Logo. Early experiments with its use have shown its value for overcoming the more difficult conceptual barriers to acquiring fluency in Logo programming.

We have developed software prototypes of two of these three types of instructional tools—an expert system/tutor, the *Algebra Workbench*, and an intelligent instructional microworld, the *Marble Bags World*—and the design specifications for a visual programming tool, the *Function Machine Laboratory*. These tools will be implemented for use in teaching mathematical inquiry skills in the areas of algebra, functions, and mathematical modeling. The content will be developed in the context of computer activities directed at building mathematical systems.

Earlier work has focused our attention on four challenges to the integration of Logo and intelligent learning tools into the mathematics curriculum. The first challenge is to develop in both teachers and students an operational understanding of the central concepts of Logo programming. This is essential for building many of the relatively complex and powerful mathematical programs of interest to beginning students. The teaching of programming in the elementary classroom at a nonsuperficial level of fluency has proved to be a difficult problem (Pea and Kurland, 1984). We hope to make a significant breakthrough in this area through the development of a unique visual programming front-end to the Logo programming language, the *Function Machine Laboratory*.

The second challenge is to move from quantitative interpretations of number in arithmetic—derived from concrete objects—to the more abstract operational concepts of variable, expression, and function in algebra and elementary mathematical analysis. The pilot study (Roberts, 1987) suggests that the *Marble Bags World* provides an extremely useful metaphor to facilitate this transition. The effectiveness of the system is due to its direct linkage of concrete iconic representations of algebraic objects and operations with the corresponding elements in the abstract symbolic language. We plan to enhance the power of the system by extending its capabilities for presenting multiple, linked representations.

The third challenge is to develop the students' ability to plan and to construct strategies. Beginning students encounter enormous difficulty coping simultaneously with the strategic issues of algebra problem solving (the decision of what goal to set next and of what algebraic operation to do to accomplish it) and the manipulative issues (the knowledge of how to do algebraic operations such as expanding terms, factoring expressions—the "busy work" of symbolic and arithmetic manipulation).

Confounding these two distinct cognitive aspects of algebra problem solving makes the task of learning algebra unnecessarily difficult and deeply discouraging. We designed an intelligent instructional system, the *Algebra Workbench*, to support students' algebraic symbol manipulation. The *Algebra Workbench* effectively separates the strategic and manipulative aspects of problem solving. It serves as the students' agent by performing the algebraic manipulations on request, thus enabling them to focus attention entirely on strategic decisions. The *Workbench* also helps students by incorporating expert programs that can be called upon to offer strategic advice, demonstrate and explain the solution of problems, or identify errors. Initial classroom work with the *Workbench* during our pilot teaching year has already established it as an extraordinarily valuable instructional tool.

The *Algebra Workbench* seems to be a productive model for the introduction of expert systems into educational software. In order to make problem solving strategies themselves the focus of the curriculum it is important to extend the scope of the *Workbench* problem domain, to develop and incorporate alternative experts, and to enable the system to present alternative solution paths to the student. We plan also to provide additional structure for exploration through the use of a problem posing expert that relies on a flexible curriculum model.

The fourth challenge is to support teachers meeting the demands placed on them for greater mathematical sophistication, more computer experience, and more flexible pedagogical style. The computer provides an exploration environment that opens many mathematical possibilities. As we alter the focus of the curriculum from computation to planning and inquiry skills, it is crucial that teachers have extensive preparation. The teacher must not only be competent in mathematics and programming, but also must be comfortable with the style of teaching that allows for structured exploration on the part of the student. We will develop intensive preservice and inservice teacher training courses very specifically targeted to these issues.

We developed the precursors to the intelligent tools and the mathematical inquiry approach proposed here in the context of teaching algebra. In the present project we will extend the work in two significant directions. First, we will expand the mathematical content of the research to extend through greater portions of the algebra curriculum, and beyond, into content areas that are not in the current curriculum: functional analysis and mathematical modeling. Second, and much more important, we have shifted the emphasis even further from the traditional curriculum focus on mathematical *content* towards a presentation focused on mathematical *process*. We will develop curriculum, intelligent tools, and support materials to introduce the teaching of the *processes* of mathematical inquiry into school mathematics.

SOFTWARE ENVIRONMENTS:
Intelligent Instructional Tools

We plan to develop three distinct types of intelligent instructional tools to support student learning activities in the mathematical inquiry modules.

i. Expert Systems and Tutors: The *Algebra Workbench*.

The *Algebra Workbench* is an example of a new class of intelligent instructional tool, distinctly different in its design and operation from current intelligent tutoring systems. It is designed to provide the student with a structured algebra exploration environment sufficiently powerful to effectively supplement (rather than supplant) the instructor. It incorporates an expert algebra symbol manipulation program capable of performing the algebraic operations requested by the student. It also includes a problem-solving specialist, an articulate tutor capable of demonstrating and explaining the planning and elaborated details of the solution process as it works through a problem (possibly posed by the student), advising students on effective strategies in the course of their work on a problem, and critiquing the students' work at the end of a session. Its instructional strategies for specifying locus of control, problem selection, and intervention, are variable and settable by the teacher or student, rather than fixed as in tutorial CAI systems.

The *Workbench* is an operational system in prototype stage. It has two parts, the Algebra Calculator and the Expert Equation Solver, which join the computer's ability to perform precise symbolic and numeric computations in real time, with artificial intelligence techniques for expert equation solving and articulate explanation of its performance along the way. The *Workbench* places the control for the interaction in the hands of the student. It allows students to take responsibility for their own learning. At the same time, its expert program is available to support the student with demonstrations, explanations, and strategic advice.

The Algebra Calculator can be used by the student to solve equations by automatically performing operations on equations or algebraic expressions. The student can perform operations (addition, subtraction, multiplication, or division) on both sides of an equation, or invoke functions ("Do Math", Remove Parentheses, Collect Terms, Expand Terms) on parts of an equation. By providing the means to perform computations, the Calculator allows the student to focus on planning strategies for solving equations.

The student can call upon the Expert Equation Solver to solve an equation. When the Expert is invoked it provides a solution path (although not necessarily the most concise one). If the student asks for advice, the Expert suggests a current subgoal, such as "eliminate fractions", or "isolate the variable". If the student asks for more advice, the Expert suggests an action that it would take, such as "multiply both sides by (n-1)". The Expert can also be invoked to do a single step, or solve the entire equation. The *Algebra Workbench* allows for multiple paths to a solution and therefore students can compare their own paths with the Expert's.

Its use in the inquiry modules requires that we extend the capabilities of the Algebra Workbench in several dimensions by:

1. Extending the algebraic range of the experts.
2. Building meta-experts with alternative solution strategies.
3. Providing alternative levels of detail.
4. Providing graphing and multiple representations.
5. Designing and implementing a Problem Selection Expert.

Extending the algebraic range. The Algebra Calculator is currently able to handle linear algebraic equations and expression simplification. We will extend the Calculator to handle inequalities and simultaneous equations. We also will extend the capabilities of the Experts to support work with nonlinear expressions and multiple variables.

Meta-expert. Currently there is one domain expert in the *Workbench*. As we extend the Calculator it is necessary to create experts for the new domains. To support the student's development of alternative problem solving strategies, we will be developing experts for different methods within a domain. This will allow the program to present different strategies to the student, and to explore the effect of different solution paths. The meta-expert will be able to examine different solution paths and present a "preferred path" according to explicit criteria. This directly addresses the mathematical thinking issues involved in strategic decisions. By showing multiple paths to a solution, we help the student to develop the mathematical aesthetic that appreciates the differences between solution paths, valuing one path for its brevity and another for its elegance.

Alternative levels of detail. The operations currently available on the Calculator are fixed at a high level of computational generality (e.g. "Expand Terms"). We will allow the student or teacher to alter the level of operations available to the student. If, for example, the teacher views *expanding terms* as a problem type (like *solving equations* in the current system) the student should be able to solve a problem of this type in terms of more basic algebraic operations (such as association or commutation operations). This "zooming in" on a step can be contrasted with the ability to "back off" and see a summary of the solution path, showing only those formulas affected when a goal has

Graphing and multiple representations. It is important to provide multiple linked representations of the same algebraic information. We will provide a graphing capability that supports graphic representations of the formulas. This opens the way for a visual introduction to analytic geometry. We will supplement the symbolic manipulation capabilities of the Calculator by introducing an iconic representation. This will help bridge students' difficulties in moving from concrete to abstract mental models.

Problem selection expert. The *Algebra Workbench* will be integrated into the curriculum through the Problem Selection Expert. The expert will have a model of the problem sequencing in the curriculum; it will be able to respond to requests from the student or teacher for problems similar to, easier, or more difficult than the current problem. The expert can be used to construct tests, generate homework problems, or construct demonstrations and exercises expressly designed to address a student's specific difficulties revealed in Algebra Workbench problem interactions.

ii. Instructional Microworlds: The *Marble Bags World*.

The second prototype instructional tool, the *Marble Bags World*, is useful for introducing the basic concepts of algebra in a concrete way through a familiar metaphor. A preliminary version of the *Marble Bags World* was implemented in Logo to enable rapid prototyping in time for its use in a pilot classroom. This system expresses algebraic operations iconically in a graphic object world.

Students in our sixth grade pilot class related easily to the metaphor of the marble bag as a concrete representation of the notion of an unknown number. The manipulations of marbles and bags in the activities of constructing and solving equations seemed very natural and clear, and the problem work proved highly engaging to students. The concept of equation and initial work in equation solving was introduced through the metaphor of the balance pan as a concrete representation of an equation. Students acquired an operational understanding of the concepts of variable and equation. Work with marble bags problems proved to be a vital bridge for transferring the concrete representation of expressions and equations to the formal algebraic notation. It helped prepare students for the more complex algebra problem work with Logo and the *Algebra Workbench*.

Extending the *Marble Bags World* We will connect the *Marble Bags World* to the *Algebra Workbench*, to create a single unified tool that will encompass the functions of both tools. In addition we will extend the scope of the system to include the representation of more general algebraic forms and the treatment of more complex problems. Two specific improvements are the implementation of "anti-marbles" and the use of dynamic grouping.

Anti-marbles. The metaphors of marble bags and the balance pan do not provide support for the algebraic operations necessary to solve problems involving negative numbers. In order to include the traditionally difficult area of signed number arithmetic we have developed the notion of *anti-marbles* representing "negative quantities". They are displayed as marbles but their icons are white (ordinary marbles are black objects). Functionally, an anti-marble cancels a marble, in the same sense that 2 and -2 cancel each other. Anti-marbles enable the iconic representation of a much greater range of problems. In the current marble bags system subtraction is an undefined operation when there are fewer marbles displayed than the number to be subtracted.

We will implement the programs for algebraic manipulation of anti-marbles and extend *Marble Bags World* to support dynamic graphic processes involving three types of objects—marbles, bags, and anti-marbles. Students manipulate the objects as before, but when a marble and anti-marble are brought into contact, they annihilate each other and disappear.

Dynamic grouping. We will extend the system to treat parenthesized expressions as "packages". Packages contain some definite number of bags, marbles, and other packages. The expression '3(4x+2)' is interpreted as "three packages, each containing 4 bags and 2 marbles." This enables problem manipulations sensitive to order of operations, or requiring factoring or distributing of expressions. We will provide a dynamic environment for manipulating the visual grouping of marbles and bags. This allows the development of activities to help students develop an intuitive feel for the effects of factoring and expanding (ungrouping) expressions.

Teacher Interface. Some students need to focus on the process of solving an equation and are overwhelmed by large numbers, while others seem to need the challenge of large numbers. The teacher interface will enable teachers to select the problems and problem modes to be presented to students, e.g. to determine whether problems would involve signed number operations, or fractional solutions. Teachers can then set parameters to constrain the characteristics of problems randomly generated by the computer or to bypass the random generation of problems entirely and enter a problem set for students. Some of the most exciting learning has come from students making up their own problems for each other to solve. A network facility that will allow students to add problems to a central databank accessible by all, will be included as well.

iii. Visual Programming: The *Function Machine Laboratory*.

Programming ideas and activities can greatly enhance the teaching and learning of mathematics. Recursive functional languages like Lisp and Logo are powerful instruments for helping students experience, understand, and enjoy mathematical ways of thinking. Acquiring a level of fluency in the use of a programming language sufficient to empower students to work on the development of interesting, mathematically rich, programs is a non-trivial task. Even Logo, the most accessible computer language, poses some significant conceptual barriers. The principal one is the understanding of the syntax and semantics of Logo functions (procedures and procedure structures which take inputs and generate an output). These are the key and crucial constructs for building the powerful mathematical programs that "take off" and do things that are interesting and exciting to students.

We will approach Logo programming in a new way that effectively bypasses these conceptual barriers. Our approach is based on new visual metaphors and iconic constructs. Our system is a visual equivalent of Logo expressly designed to make the hard parts of Logo more accessible. Program development will involve something like the building of complexes of "machines" tied together in the manner of a Rube Goldberg machine. It is an advance over existing visual programming systems such as *DataFlow*, *STELLA*, and *Show and Tell* which have limited scope of application. These systems are specialized to particular problem domains and do not support a fully general programming language.

Our iconic programming language is called the *Function Machine Laboratory*. A student uses the laboratory to design and build complex machines out of standard parts (simpler machines). A machine is the visual equivalent of a Logo procedure. Machines can have inputs and outputs—they are isomorphs of the function concept in mathematics. The main building block, called the *processor*, is a visual representation of a machine that requires inputs and that produces an output, a *function machine*. It has *hoppers* for its inputs (at the top) and it has an output *spout* (coming from the bottom). A *source* machine has an output but no inputs; a *destination* machine has one or more inputs but no output; and a *solo* has neither output nor inputs. Machines and inputs have names.

In our metaphor, each machine is thought of as a Logo procedure. The body of a machine consists of Logo instructions. The machines are linked to form more complex structures by connecting the output of a source or processor to an input of a destination or processor. A complex machine can be represented by a single icon; this process can be extended recursively so that very complex machines with several levels of structure can be shown at the top level as a single icon. One can also open any complex machine to show its internal machine structure.

The *Function Machine Laboratory* is a visual language with the full representational power of a universal programming language. There is essential equivalence between mathematical functions, Logo programs, and function machines. But the function machine diagrams show the structure of complex programs and functions more transparently than either of the symbolic representations. They help the student see what parts are necessary to complete a structure and how the parts must communicate. These capabilities have direct relevance to applying functions to solve problems.

This is a three year project beginning in late Fall, 1987. The first part of year one will be spent principally on software development. Our presentation will focus on our progress with the software environments: the *Algebra Workbench*, the *Marble Bags Lab*, and the *Function Machine Lab*.

REFERENCES

Pea. R.D. & Kurland, D.M. (1984). On the Cognitive Effects of Learning Computer Programming, *New Ideas in Psychology*, 2, 137-168.

Polya, G. (1962) *Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving*. New York: John Wiley.

Roberts, Nancy, Richard Carter, Frank Davis, Wallace Feurzeig, Power Tools for Teaching Algebra, Final Report to the National Science Foundation, Lesley College, 1987

GUIDED PROBLEM SOLVING, YOUNG CHILDREN, LOGO, AND IMPROVED HIGHER ORDER THINKING SKILLS

Ann Shore
1290 Ridgeview Way
Bonita, CA 92002
(619) 421-8516

In this paper arguments will be presented for the use of Logo as a means of improving higher order thinking skills in young children. It will be further argued that this goal can best be achieved in an educational environment that is greatly dissimilar to that espoused by Papert. This paper will investigate the hypothesis that, contrary to the traditional unstructured, discovery method of Logo use, young children's higher order thinking skills will most effectively be enhanced through direct instruction and guided problem solving in Logo use.

When young children are posed with a problem, they typically accept the first solution that presents itself. If we accept that problem solving skill is an important life long skill and that divergent thinking is a vital aspect of problem solving, then divergent thinking. . . . "should be encouraged at an early age so that it becomes a natural and accepted part of children's intellectual functioning" (Cliatt, Shaw, and Sherwood, 1980, p.1063). Problem solving skills should be analyzed and presented at varying levels of difficulty in a developmental fashion. Experiences in divergent problem solving should be an early and ongoing portion of the child's school life.

Computers offer unique problem solving opportunities for students. The flexibility of the computer permits students to propose, discover, and test out alternative strategies to solving problems (divergent thinking) . Use of the computer as a tool in problem solving provides the student with valuable decision making experiences.

The use of Logo on the computer can provide educational experiences possible only through computer use. It can provide an environment conducive to meaningful learning about problem solving for a wide variety of ability levels, much more so than most traditional approaches to instruction. Logo is ideally suited to foster improved skill in the planning stage of problem solving because the flexibilty to produce alternate solutions is an integral portion of Logo use.

Not only is Logo thought to have potential for improving problem solving skills, there have also been assertions of its ability to increase higher order thinking skills (O'Brien, 1983; Pogrow and Buchanan, 1985). Papert (1980b) has suggested that the use of Logo may be an appropriate means of providing a challenging, motivating environment that is individually paced; thus, perhaps producing enhanced and more efficient cognitive development.

LOGO AND YOUNG CHILDREN

Logo provides access for young children (ages 4-7) to problem solving experiences on the computer (Lawler, 1985; Williams and Williams, 1984). Through Logo use, young children can be free to explore and create without becoming frustrated by a lack of skill or fine motor ability. When using Logo, the pictures children create are the problems they have set for themselves; and therefore, have a psychological investment in discovering the solution (Clements, 1985a; Lawler, 1985; O'Malley, 1985). Through problem solving experiences on the computer, young children may be activating cognitive skills and strategies previously thought too advanced for their developmental age (Clements, 1985b; Clements and Gullo, 1984; Papert, 1980). There is some evidence that Logo programming experiences may actually accelerate progress through the Piagetian stages of development (Clements, 1985a; Lawler, 1985; Williams and Williams, 1984).

Although Logo is a very accessible computer language, research has indicated that young users (ages 3-6) can find the process difficult and complex (Pea, Hawkins and Sheingold, 1983). There are some obstacles which must be overcome when providing Logo programming experiences for young children: (a) The task of remembering how to correctly spell the necessary input words may be too difficult, (b) the action of actually typing may be too complex for the fine-motor skills of young children, (c) young children may not have sufficient mathematical skill in estimating distances and angles, and (d) the task of procedurally ordering a sequence of "turtle" moves without immediate feedback after each move may be beyond the ability of young children (Clements, 1983).

There are, however, means of surmounting these obstacles. Given the proper support and assistance, even pre-school children can have successful programming experiences and can demonstrate creative logical thinking. This support can be in the form of simple, programmable toys such as "Big Track" (Keller and Shanahan, 1983), elaborate robots (Perlman, 1976), or modifications of the Logo language to allow for easier input and more instantaneous feedback (Clements, 1983).

LOGO AND DISCOVERY LEARNING

The rationale behind discovery learning is that concepts discovered by an individual become an integral, functional portion of a personal knowledge base which can be used to solve other problems effectively (Bruner, 1966). Discovery learning is grounded in the theory that the child's cognitive and intellectual development is the result of active, rather than passive, interaction with the surrounding environment (Inhelder and Piaget, 1958). For many Piagetians, the advent of the computer has broadened exploration potential for children in a manner and scale never before possible (Lawler, 1984; Rheingold, 1983; Williams and Williams, 1984).

Papert (1984b), while strongly convinced that children must make their own discoveries in order for cognitive development to occur, believes that the level of cognition is a direct result of the type of environment provided. He is supported in this belief by others who believe similarly, that cognitive development is directly linked to the environment provided (Bruner, 1966; Gagne, 1977). It is proposed that by allowing young children to freely explore the world of Logo problem solving, with little or no instructional intervention, that passage through stages of cognitive development may be accelerated (Papert, 1984b; Williams and Williams, 1984).

LOGO AND DIRECT INSTRUCTION

The concern of many researchers and educators in regard to the discovery mode of instruction is that, left on their own to discover higher order thinking and problem solving strategies, many children may never open the right doors to do so; and, if do develop sophisticated problem solving skills, may never generalize or transfer these skills to other areas (Donaldson, 1978; Gallini, 1985; Leron, 1984). There is also some indication that the discovery method is not appropriate for young children because of short attention spans, inability to delay gratification, and a lack of understanding of what a problem or a solution is (Bruner, 1966).

While the world of Logo programming may encourage the development of problem solving and higher order thinking skills, computers and programming languages by themselves will not generate a world of better thinkers (Pogrow and Buchanan, 1985). Donaldson (1978) has stated, "If intellectual powers are to develop the child must gain a measure of control over his own thinking and he can not control it while he remains unaware of it" (p. 129). Direct instructional intervention must be provided to make students aware of the thinking processes utilized and to enhance the potential for transfer of problem solving skills to other areas (Clements, 1985a; Leron, 1984).

Once students have a framework for organizing thinking, Logo can be the means for reflection on and practicing of divergent problem solving skills (Leron, 1984). Logo use for problem solving can then become an avenue of discovery as students, in an organized fashion, analyze components, synthesize findings and evaluate results (Leron, 1984; Perkins, 1986). As Gagne (1966) has stated "the strategies of discovery...do not have to be learned by discovery" (p.148).

LOGO AND GUIDED PROBLEM SOLVING

The task then is to provide an educational environment that accesses the child's pre-existing cognitive structures and enables the expansion of those structures in an efficient and creative manner while maintaining awareness of individual needs and capabilities. This study is an investigation into the relative effectiveness of two different Logo programming experiences--one which provided guidance in problem solving skills and one which provided time and opportunity to randomly explore Logo--in the enhancement of young children's higher order thinking skills.

PROJECT DESCRIPTION

The subjects of this study were 62 five to six year old students: 48 kindergarten students and 14 first grade students. They were members of a kindergarten - first grade combination class and an afternoon kindergarten class. The study took place at a public elementary school (K-6) in a suburban community which served approximately 450 students.

This research study was a pretest-posttest control group design which took place over a four month period in the second half of the school year. Students were randomly assigned to one of three experimental groupings: two Logo programming groups (Guided Problem Solving and Discovery Mode) and one control group. Members of the Logo groups were assigned partners to work with while using the computer. This was done for two reasons: (a) to increase the potential for student/student verbal interactions while using the computer and, (b) to increase access time for computer use. There were approximately twenty students in each of the two treatment groups and one control group.

During the course of the study, the students of the experimental treatment groups were attempting to find alternate means of solving the same problems--how to use Logo commands to reproduce paper and pencil drawings on the computer--one group through incidental discovery (DM) and one group through guided problem solving (GPS). A variation of the Apple Logo programming language was employed. The "Zoom" variation provides for single keystroke input which results in instant cursor movement (Clements, 1983).

All students received instruction in basic keyboarding skills. A game-like format for practice in finding specific letters on the computer keyboard was implemented. The students worked on Apple IIe computers.

SPECIFIC TREATMENT DESCRIPTIONS

Off-Computer Activities: Both Logo groups--Guided Problem Solving and Discovery Mode (GPS and DM)--participated in three weeks of preparatory off-computer activities and lessons. These activiites consisted of analyzing the difference between turns and forward or backward movement, participating in "robot" simulations, and moving a toy turtle through increasingly difficult mazes in response to specific commands.

Guided Practice Computer Activities: Both Logo groups were then involved in four weeks of guided practice activites. Each week one or two new Logo commands were introduced. While these introductory Logo activities were being taught and practiced, both the GPS and the DM student teams were provided assistance and guidance in the analysis of appropriate Logo input to achieve the desired results. At all times, students were encouraged to use their computer partner as the first source for assistance when a problem arose (Clements, 1985b). Much emphasis was placed on discussing the current computer task with the partners and arriving at mutually agreed upon solutions or plans of action (Johnson, Johnson and Stanne, 1986).

Experimental Treatment Variations: After seven weeks of instruction and guided practice, most of the students demonstrated a basic understanding of Zoom Logo functions. The next task was for each student to produce a drawing on a piece of paper cut to the same size as the monitor screen. Each team then had to decide which drawing it wanted to reproduce on the computer. From this point on, separate instructional sessions were held for each treatment group.

Discovery Mode: The members of the DM group were directed to talk to their partners to figure out how to tell the "turtle" to draw the decided upon picture. No other assistance was provided except for an occasional reminder of the function of command keys or technical assistance if the program malfunctioned (Pogrow and Buchanan, 1985). Questions of a technical nature were answered, but all other queries were met with, "Ask your partner." or "Talk to your partner." DM students were frequently encouraged to work together on the computer.

Guided Problem Solving: The focus of instruction for the GPS group was to provide guidance and practice in the identification of a problem, analysis of alternative plans of action, and the evaluation of results (Gallini, 1985). Students of the GPS group were given lessons in how to determine the specific "turtle" moves needed to produce the picture selected. Instruction was provided in the production of written plans. GPS students met as a group on a regular basis to compare, contrast and discuss the plans and results of each GPS team.

TEST MEASURES AND DATA COLLECTION

Very little classroom-based research has been conducted concerning kindergarten and first grade students, the use of Logo, and improved cognitive strategies. This is partially attributable to the difficulty of measuring problem solving and higher order thinking skills because of a lack of standardized test measures suitable for the five to seven year old age group. This study attempts to circumvent this difficulty through the use of alternate test measures and through the extensive use of video tapes. A Puzzle test was developed by the researcher as a measure of divergent thinking. The "Torrance Tests of Creative Thinking, Figural Form A" (Torrance and Ball, 1984) were selected as an additional measure of the skills under investigation.

Higher Order Thinking Skills and Task Persistence. Two days a week video tape recordings were made of each team as they worked on the computer. The video tapes made possible later item analysis of targeted indicators of higher order thinking skills. Using Bloom's (1956) *Taxonomy of Educational Objectives: Cognitive Domain* as a guideline, three categories of cognitive development were selected: Applications, Analysis, and Evaluation.

Since task persistence is also a measure of divergent problem solving ability, the video tapes also provided a means of comparing time-on-task behavior between the two experimental treatment groups (Gowan, 1967; Guilford, 1967; Torrance, 1962; Wells, 1984). Task persistence or time-on-task was measured by computing the percentage of time during each computer session that both members of the team were actively engaged in solving the problem at hand.

PROCEDURES

Three Apple IIe computers with color monitors were placed in a small workroom adjacent to the kindergarten classroom. All computer work and instruction was done in this secluded area to prevent contamination between treatment groups. The computer groups (GPS and DM) worked at the computers on a regular basis twice a week for about 15 to 20 minutes each day. On any given day, only members of a particular treatment group worked in the computer room.

In order to prevent the students in the control group from feeling completely excluded from computer use, approximately every four to five weeks, a day was set aside as game day. Computer games involving the matching of letters, beginning sounds and nursery rhymes were set up on the computers. All students, regardless of experimental grouping, participated in playing the games. It was felt that the benefits of keeping peace within the classroom and maintaining positive relations with parents far out weighed the minimal exposure to computer use by the control group. The games played were unrelated to the variables under study--divergent problem solving and higher order thinking skills. It was felt that this exposure would not contaminate the end results of the study.

DISCUSSION

The hypothesis that the Logo using groups taken as a whole would demonstrate greater effectiveness in divergent problem solving skills than the control group was supported by the Puzzle test data, $F(2,54) = 3.37$, $p < .05$; but not, however, by the data from the Torrance Tests of Creative Thinking, $F(2,54) = 1.10$, $p > .10$. An inspection of group means on each of the Torrance sub-tests of divergent thinking indicated some possible differences between group means on the sub-test of Elaboration and the sub-test of Resistance to Premature Closure. These two sub-tests are most closely linked with task persistence, a characteristic of divergent problem solving. These differences, however, were not deemed significant upon statistical analysis, $F(2,54) = 2.79$, $p > .05$. This somewhat typical research problem is a result of means cancelling each other. Salomon and Gardner (1986) have suggested that the inability to discover significant differences when comparing group means becomes especially relevant when applied to research on computers involving open-ended, interactive, self-directed learning which is likely to produce divergent results.

The most impressive findings were revealed in the analysis of the videotapes. Over 60 hours of videotapes were viewed in random order, analyzed and scored for task persistence, indicators of higher order thinking and general comments. Both the hypotheses that the GPS group would demonstrate significantly greater effectiveness in problem solving skills and in higher order thinking skills than the DM group were strongly supported by the data compiled from the item analysis of the video tapes. Especially noteworthy were the findings on task persistence, $t(28) = 32.81$, $p < .001$ and on Analysis indicators, $t(28) = 22.60$, $p < .001$. The self-directed discovery experiences in Logo programming by the DM group not only did not result in improved cognitive strategies, but also original competency levels in both areas were actually reduced (See Table 1).

In an effort to further understand the really dramatic findings in relation to task persistence, the videotapes were again viewed--this time in chronological order and by treatment group. By examining the tapes in this fashion, very clear, distinct patterns of behavior between the Logo groups emerged. During the first weeks of instruction in Logo use, it was very difficult to distinguish between treatment groups. Members of both groups spent approximately equal amounts of time on and off task, distracting each other and being distracted by each other or anything else that happened to move in the workroom. This included clowning around in front of the video camera. One student was finally denied access to the computers because his behavior was so disruptive. Any data collected on this student was not included in the statistical analyses.

The differences between groups began to emerge at about the fifth and sixth weeks of computer use--the point at which the GPS group began developing written plans. The students in the GPS group were now observed to be talking to each other about their projects, analyzing potential "turtle" moves and, above all, planning. Meanwhile, the members of the DM group were functioning in pretty much the same manner as they began.

As the weeks progressed, it became very clear that the members of the DM group rarely, if ever, settled upon a mutually agreed upon task--never identified a problem. Despite frequent reminders to work together, the typical mode of operation amongst members of the DM group was for one team member to work at randomly inputing "turtle" moves while the other student waited patiently, or not, for a turn. When the second student got tired of waiting for a turn or was granted a turn by the first user, the most common action was to erase all the work of the previous user. The one or two DM teams who did identify a problem and mutually agreed to target a specific picture for Logo input on the computer were rarely sucessful in their efforts. Because they were unable to discover an organized method for solving their problem, any original efforts at cooperation rapidly fell apart. Even though the DM students were constantly reminded to work together to select and solve a problem, without a framework or guidelines for how to proceed; the DM students, not only were unable to achieve successful programming experiences, but were also unable to discover how to work together to identify and solve a mutually decided upon problem.

It is important to note that no matter to which Logo treatment group a student belonged, inattention to task was not related to a lack of understanding of Logo commands or their use. Even the youngest, most distractible student demonstrated little or no difficulty with Logo language or with "decentering"--a cognitive skill which Piaget attributes to the stage of formal operations. These students were five to seven year-olds, supposedly in the preoperational stage according to Piaget's guidelines (Williams and Williams, 1984). The discovery that young children can "decenter" provides credence to Donaldson's (1978) premise that children can perform at a higher level of cognitive skill when provided an appropriately relevant framework for functioning. The DM group had learned the functional use of the Logo language as well as the GPS group but had not learned problem solving strategies.

RECOMMENDATIONS

The use of the computer in the classroom is not a passing technological marvel soon to be relegated to the educational sidelines along with the television and the overhead projector--technological marvels of the past. Evidence of this is that educators are every day discovering new and more creative ways to make use of the computer in the classroom. In fact, computer use in the classroom has blessed the television with a resurrgence of use--as a large screen monitor for whole class instruction in computer use. Meanwhile, computer technology is changing so rapidly that new, more powerful, easier to use, and less expensive computers are becoming available on the market.

The acquisition of problem solving and higher order thinking skills is too important to be left to chance. A critical element in the successful enhancement of the higher order thinking skills of the GPS group was teacher intervention. The instruction and guidance in problem solving strategies provided by the teacher were the deciding factors in this success. It was the process that was being taught that was the important component. The combination of the computer and Logo programming enabled kinergarten and first grade students to focus, at an entry level, on specific problem solving strategies without obscuring the targeted skill by the imposition of traditional curriculum content.

It is vitally important that teachers, new and experienced, also have instruction and guidance in the teaching strategies most likely to encourage the improvement of problem solving and higher order thinking skills. Key features that contributed to success in this study were:

1. Specific problem solving strategies were isolated, modeled, and practiced in group settings.
2. An emphasis was placed on cooperative learning. Students worked in teams to accomplish a mutual goal rather than in competitive isolation. Cooperative interpersonal relationships were modeled and encouraged.
3. A common framework and vocabulary for discussion about problem solving was provided.

It is strongly reccommended that future use of Logo in the primary classroom not be taught in isolation as was the case in this research project. It is postulated that the cognitive skills gained through Logo use in problem solving would have been greatly enhanced through direct instruction in the application of those same skills for problem solving in the regular classroom curriculum. Although it is appropriate to initiate specific cognitive skills instruction in isolation, as soon as is possible, the use and practice of these skills must be integrated throughout the curriculum to ensure their transfer to other areas.

As educators, we have a responsibility to ensure that the instruction and use of higher order thinking and problem solving skills be provided for all students, no matter their age or ability, and not to just a few of the intellectual elite. As researchers and educators, we have the task of investigating the ways and means of implementing successful teaching strategies for the development of cognitive skills in the regular classroom. The use of the computer, combined with the programming language Logo and instruction in problem solving strategies, has proven to be a successful tool in the teaching of these skills, even at primary grade levels.

REFERENCES

Bloom, B. S. (1974). *The taxonomy of educational objectives: Affective and cognitive domain*. New York: David McKay Company.

Bruner, J. S. (1966). Some elements of discovery. In L. S. Shulman & E. R. Keislar (Eds.), *Learning by discovery: A critical appraisal* (pp. 101-114). Chicago: Rand McNally.

Clements, D. H. (1983). Supporting young children's Logo programming. *The Computing Teacher*. 11(5), 24-30.

Clements, D. H. (1985a). *Computers in early and primary education*. New Jersey: Prentice Hall.

Clements, D. H. (1985b, April). Differential effects of computer programming (Logo) and computer-assisted instruction on executive processes and cognitive development. Paper presented at the meeting of the Society for Research in Child Development, Toronto.

Clements, D. H. & Gullo, D. F. (1984, December). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76, 1051-1058.

Donaldson, M. (1978). *Children's minds*. New York: W. W. Norton & Co.

Duckworth, E. (1979). Either we're too early and they can't or we're too late and they already have! *Harvard Educational Review*, 49, 297-313.

Gagne, R. M. (1966). Varieties of learning and the concept of discovery. In L. S. Shulman & E. R. Keislar (Eds.), *Learning by discovery: A critical appraisal* (pp. 135-150). Chicago: Rand McNally.

Gagne, R. M. (1977). *Conditions of learning* (3rd. ed.). New York: Holt, Rinehart and Winston.

Gallini, J. K. (1983, April). What computer-assisted instruction can offer toward the encouragement of creative thinking. *Educational Technology*, 23(4), 7-11.

Gowan, J. C. (1967). What makes a gifted child creative: Four theories. In J. C. Gowan, G. D. Demos & E. P. Torrance (Eds.), *Creativity: Its educational implications* (pp. 9-15). New York: John Wiley and Sons.

Guilford, J. P. (1967b). Potentiality for creativity. In J. C. Gowan, G. D. Demos & E. P. Torrance (Eds.), *Creativity: Its educational implications* (pp. 83-88). New York: John Wiley and Sons.

Inhelder, B. & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. New York: Basic Books.

Johnson, R. T., Johnson, D. W., & Stanne, M. B. (1986, Fall). Comparison of computer-assisted cooperative, competitive, and individualistic learning. *American Educational Research Journal*, 23, 382-392.

Keller, J. & Shanahan, D. (1983, May). Robots in the kindergarten. *The Computing Teacher*, 10(9), 66-67.

Lawler, R. W. (1984). Designing computer-based microworlds. In M. Yazdani (Ed.), *New Horizons in Educational Computing* (pp. 40-53). Chichester, England: Ellis Harwood, Ltd.

Leron, U. (1984, Fall). Quasi-Piagetian learning in Logo. *Journal of Computers in Mathematics and Science Teaching*, 4(1), 52-53.

Norris, S. (1985 May). Synthesis of research on critical thinking. *Educational Leadership*, 42(8), 40-45.

O'Brien, T. (1983, October). Five essays on computers in education. *Phi Delta Kappan*, 65(2), 110-112.

O'Malley, C. (1985, March). Boosting your child's creativity. *Personal Computing*, 9(3), 100-107.

Papert, S. (1980a) *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.

Papert, S. (1980b). Teaching children thinking. In R. P. Taylor (Ed.), *The Computer in the School: Tutor, Tool, Tutee* (pp. 161-176). New York: Teachers College Press.

Pea, R., Hawkins, J. & Sheingold, K. (1983, April). Developmental studies on learning Logo computer programming. Abstracts from the Biennial Meeting of the Society for the Society for Research in Child Development (Summary).

Perlman, R. (1976). Using computer technology to provide a creative learning environment for pre-school children. (Logo Memo No.24). Cambridge, MA: M. I. T. Artificial Intelligence Laboratory.

Pogrow, S. & Buchanan, B. (1985, September). Higher-order thinking for compensatory students. *Educational Leadership*, 43(1), 40-43.

Rogers, L. M., & de Leeuw, G. J. (1985). Imagery, language, Logo and curriculum. Unpublished manuscript, University of Calgary, Faculty of Education, Calgary, Canada.

Salomon, G. & Gardner, H. (1986, January). The computer as educator: Lessons from television research. *Educational Researcher*, 15(1), 13-19.

Torrance, E. P. (1962). *Guiding creative talent*, Englewood Cliffs, NJ: Prentice-Hall.

Torrance, E. & Ball, O. (1984). *Torrance tests of creative thinking: Figural A*. Bensenville, IL: Scholastic Testing Service, Inc.

Wells, G. (1984). The relationship between the processes involved in problem solving and the processes involved in computer programming. In *Capitalizing on computers in education. Proceedings of the 1984 Association for Educational Data Systems Annual Conference*, MA.

Williams, F. & Williams, V. (1984). *Microcomputers in elementary education.* Belmont, CA: Wadsworth Publishing.

TABLE 1

Task Persistence and Analysis Indicators

	Guided Problem Solving		Discovery Mode	
	M	SD	M	SD
Analysis				
Pre Test	3.57	2.71	7.12	6.14
Post Test	26.29	16.31	1.75	2.29
Task Persistence				
Pre Test	54.12	9.93	63.86	22.68
Post Test	89.88	7.38	48.43	17.14

EFFECTIVE STAFF DEVELOPMENT: PREPARING TEACHERS TO USE COMPUTER-AS-TOOL

Leslie Conery
Vivian Johnson
David Moursund
Billy C. Yates
Center for Advanced Technology in Education
University of Oregon
Eugene, OR 97403

ABSTRACT

The ultimate goal of all staff development is to improve the quality of education received by students. This panel focuses on staff development for use of computer-as-tool in schools. This is a particularly challenging area since few teachers learned about computers while they were in school, and the computer field is still changing quite rapidly. All four panelists are involved in research and implementation of staff development for instructional use of computers, and all are involved in a National Science Foundation project that is producing staff development materials for use of computer-as-tool in schools. These materials are being published by the International Council for Computers in Education.

Certain aspects of staff development transcend all potential audiences, while others are quite dependent on the grade level and subject matter orientation of the participants. For example, we know that in most situations "one shot" inservices are not very effective, and we know that the staff development needs of secondary school science teachers are quite a bit different from those of elementary school teachers. This panel will cover both general ideas and some specifics for particular groups of educators.

Elementary school teachers are accustomed to operating within a multidisciplinary environment. A day's lesson plans may cover ten or more distinct disciplines, and the curriculum is over-crowded. Moreover, the number of computers available in elementary schools is still severely limited. It is a major challenge to help elementary school teachers learn enough about computers to begin to integrate their use into their multidisciplinary classrooms. Leslie Conery will address this topic.

One major and unifying theme in secondary school science is the generation and testing of hypotheses. An integrated software package, combining spreadsheet, database, graphics, and word processor, provides a tool that can make a substantial contribution to science education. Computer-oriented staff development for secondary school science teachers needs to be carefully balanced between hypothesis generation and testing, and use of computer-as-tool to aid in this and related endeavors such as data gathering and control of experiments. Vivian Johnson will address this topic.

Secondary school mathematics teachers feel locked into a rigid curriculum. The "lock" comes from a combination of standardized testing, courses arranged in a vertical prerequisite structure, and quite rigid scope and sequence specifications. Although computers can solve or help solve most of the types of problems covered in secondary school mathematics, teachers do not see how to implement major computer-related changes. A primary goal of staff development in this area is to help teachers overcome the initial hurdle of getting started in helping their students make routine use of basic tools such as computer graphics and equation solvers. David Moursund will address this topic.

The major goal of secondary school social studies is to prepare students for effective participation in society. Students need to gain an awareness of events and forces in the world that influence local, national, and international situations. Critical thinking, or problem solving in a broad sense, is a unifying theme in social studies education. Secondary school social studies teachers need to deal with computers as an aid to information storage and retrieval, graphics representation, modeling and simulation, and telecommunication. They also need to learn to make instructional use of computer simulations, both in whole class and in smaller group settings. Billy C. Yates will address this topic.

MUSIC, MIDI, AND COMPUTERS

Don Latarski
Adjunct Faculty, Jazz Guitar Studies
University of Oregon School of Music
Eugene, OR

ABSTRACT

The expansion of the personal computer into the field of music has brought about a revolution in the way music is made. This revolution was made possible by the standardization of a universally accepted mode of communication between computers and electronic devices such as drum machines, synthesizers, mixing boards, and signal processing equipment. This mode of communication is called MIDI (Musical Instrument Digital Interface). My presentation will begin with a brief overview of the often confusing relationship between MIDI, computers and music making devices.

The focus of my presentation will be on the types of software and hardware available to the amateur, professional and teacher of music. I'll talk and demonstrate the power of a small MIDI system with the use of a synthesizer, drum machine and MIDI equipped guitar controller and computer. (I'll be working with the Macintosh computer in all of these demonstrations.)

Convincing music can now be made by would-be musicians, made possible by the advent of MIDI. MIDI gives a person exacting control over the tools of sound production by reducing all musical events to a series of 0's and 1's. Once musical events are in the binary domain, word processing-like activities, such as cut and paste, can be performed on the musical data. But this is only the beginning. More complex activities, such as transposition, tempo shifting, note quantization (for exact metric precision), pitch bending, note by note editing and time shifting, can also be easily executed. MIDI is capable of storing every nuance of the performer while simultaneously offering the unprecedented opportunity for microscopic editing of these nuances. The choices now offered to musicians and non musicians alike are nothing short of mind boggling.

MIDI is not all bliss. With the increased control over the compositional and editing processes, comes an ever increasing responsibility on the part of the musician to broaden his or her understanding of computers, synthesizers, MIDI controllers and the myriad of related hardware and software. One can no longer be content to just play an instrument. It's hardly enough in our time, although certainly as valid as ever. Training in computers and MIDI will soon be a required part of all music students education and has already become a part of the standard curricula at many of our leading music schools.

MIDI is leading to specialization in the music field. It has fostered the rise of people who specialize in different aspects of MIDI, such as synthesizer programming, software development, related hardware development, systems experts (IBM & clones and Apple), even individual manufacturer specialists, (Yamaha, Roland, New England Digital and Fairlight). The range, variety and depth of the field is staggering and growing by leaps and bounds. Witness the depth of the electronic instrument offerings by some of the above named companies.

MIDI is now being used extensively in music composition (sequencing programs and artificial intelligence programs), ear training, fun & entertainment, teaching music reading, teaching jazz improvization, music printing, patch editing and program librarians for creating and storing synthesizer data, sampler editing programs, rhythm machine programming, transferring bulk data dumps and system exclusive data, for actual digital recording, as a link in the synchronization of tape recorders, computers and video equipment, and the list goes on and on. There's hardly an area of music which has not been radically affected by MIDI. It's safe to say that there's a software package out there to help you do whatever music related task you want. If not, there soon will be if there's sufficient demand.

Initially, MIDI was most accessible to the keyboardist. It was designed with the keyboardist in mind. Over the past several years, the power of MIDI has opened up to all other types of instrumentalists and vocalists. There are now MIDI controllers, MIDI devices which generate the approprite stream of MIDI data, for nearly everyone. You can now harness the power of MIDI from a guitar, a saxophone, violin, drum, bass guitar, or voice. The rise of these new controller devices has really fueled the industry, making possible ever expanding markets for there synthesizers, drum machines, signal processors (reverb, equalization, digital delays, etc.), drum pads, sequencers, software and interface boxes.

The danger of MIDI is subtle yet real. The manufacturers would have us believe that anyone can now make wonderful music simply by acquiring the appropriate technology. While this may be true for some, it is generally false for most. MIDI is a powerful tool, but nothing more. In the hands of a person of innate and insatiable curiosity (and lots of free time), great music can be had. In the hands of an inexperienced novice, great music is rarely made. MIDI has given us a better paint brush. However, it's up to each person to paint their own picture.

UNIVERSITY MICROCOMPUTER APPLICATIONS: SCIENCE/HEALTH

James Ellis
Ernesto A. Randolfi, Jackie L. Lloyd
Michael R. Vitale, Nancy Romance

ABSTRACT: A Model for Training Science Teachers to Use Microcomputers

James D. Ellis, Senior Staff Associate, Biological Sciences Curriculum Study, Colorado Springs, CO

The Biological Sciences Curriculum Study (BSCS), with support from the National Science Foundation, is conducting a three-year project to develop a model for training teachers to use microcomputers to teach school science (K-12). The goals for the project are to:

1. Develop and test a model of implementing educational computing in school science;
2. Train 260 science teachers and administrators in the Pikes Peak region of Colorado to use microcomputers to inhance science learning and teaching;
3. Establish a network in the Pikes Peak region to implement educational computing in school science; and
4. Disseminate a model of implementation for educational computing in school science.

The major focus of the project is to design, test, and disseminate a model for implementing educational computing in school science (K-12). During the first year, project staff and members of the advisory committee designed strategies for training science teachers and for supporting the implementation of educational computing. Project staff conducted a two-day workshop to introduce the knowledge and skills needed for integrating educational computing into science instruction. Participating teachers attended four seminars during the school year to practice and refine the skills. Throughout the year, project staff coached the participants as the participants began using microcomputers.

The project met its first-year objectives and was successful at increasing science teachers' use of microcomputers. Project staff gathered descriptive data from the participants to determine background characteristics, prior experience with microcomputers, and prior education in science and science education. Project staff defined and measured implementation according to the Concerns Based Adoption Model (CBAM). Results from pretests and posttests using the Stages of Concern questionnaire indicated that the participants changed from a profile typical of nonusers of an innovation toward one typical of users. Project staff developed an Innovation Configuration checklist to describe participants' use of microcomputers. Most of the participants were using microcomputers several ways by the end of the year. The participants identified barriers to implementing educational computing in science instruction, which project staff used to design implementation strategies for the second year of the project.

ABSTRACT: Computer Assisted Instruction in Human Sexuality

Ernesto A. Randolfi, PhD, and Jackie L. Lloyd, BS, Department of Health and Physical Education, Syracuse University, Syracuse, NY 13244

A review of the research on computer-based sex education reveals that computers are a desirable medium for instruction in this area of health education. Since both teachers and students may be uncomfortable discussing aspects of sexuality, computer-based sex education provides instructors with a unique opportunity to accurately and objectively deal with a student's most personal concerns.

A comparative analysis of microcomputer software packages in the area of human sexuality was conducted to identify common trends in computer-assisted sex education and to offer suggestions for program developers. This study was also implemented to provide health educators with a consumer-based analysis of commercially produced software for sex education. Software packages were assessed for content inclusion, accuracy of information, age level appropriateness, educational strategies, motivational techniques, and technical characteristics.

Buyers of computer-assisted health education must be concerned with content accuracy and appropriateness in addition to instructional characteristics. This presentation includes a discussion of the study's findings with recommendations for software developers and consumers.

ABSTRACT: Using Artificial Intelligence Environments to Enhance Elementary Science Instruction

Michael R. Vitale and Nancy Romance, Center for Educational Technology, Florida Atlantic University

PROLOG and expert systems shells provide unique learning tools through which teachers and students can gain increased science understanding by constructing operational logic-based models to explore and apply abstract knowledge and concept relationships. This presentation provides a status report of a continuing project designed to explore the use of PROLOG and simple expert systems within the elementary school science curriculum and in the teacher preparation process.

In this project, undergraduate preservice teachers will learn to use open-ended microcomputer-based artificial intelligence environments (PROLOG, expert systems) as tools to enhance their understanding of declarative (content) and procedural (applications) forms of science knowledge. In doing so, declarative knowledge of elementary school science concepts and relationships elaborated graphically as semantic nets will be incorporated into simple logic programming activities using PROLOG. Similarly, procedural science knowledge from the same domain will be used to construct simple rule-based expert systems focusing upon the application of science principles.

An example of a logic programming activity would be representing stages and major causal factors of the water cycle as a declarative PROLOG program and then querying the system for implications as different causal conditions are modified. A complementary procedural application using an expert system shell would be to use weather principles to construct a simple weather predictor that would assist with the interpretation of newspaper weather maps. In complementary activities, teachers also will learn to modify their preservice learning tasks for future use as instructional activities with upper elementary students in regular classroom settings. A replicated 2-group (treatment vs control) research design will determine the immediate and follow-up effect of the preservice program upon knowledge of science concepts, related science teaching attitudes, and during subsequent field-experience, classroom performance ratings. In coordination with the preservice program, the project will develop prototype microcomputer-based classroom science activities using PROLOG and expert systems with fourth-sixth grade laboratory school students.

SONGS FROM THE AGE OF CLASSROOM COMPUTING

Don Rawitsch
MECC
St. Paul, MN

Peter Rawitsch
Glenmont Public Schools
Glenmont, NY

ABSTRACT

Research and analysis is one way of expressing our knowledge of educational technology. Satire and music provides an alternative perspective. Don and Peter Rawitsch have enhanced their teaching and training over the past ten years with original songs documenting the excitement and frustration that has accompanied the growth of classroom computing.

This performance consists of a variety of songs written and sung by the brothers that chronicle such topics as the trials of implementing computing in schools; the challenge of serving as a local computing coordinator; teaching the Logo language to young children; and the impact of computing on teachers.

The growth of classroom computing has been accompanied by the publication of books and magazines. Very few songs, however, have been written about the field. Most that do exist will be performed here!

EVALUATION OF LOCAL AND REGIONAL TECHNOLOGY PROGRAMS AND POLICIES

Betty Collis
Chair and President, ICCE
Department of Psychological Foundations
University of Victoria
Victoria, BC, Canada

Gail Marshall
Adjunct Professor
Department of Behavioral Studies
University of Missouri-St. Louis
St. Louis, MO

Sheila Cory
Chapel Hill-Carrboro City Schools
Lincoln Center/Merrittvill Rd.
Chapel Hill, NC

ABSTRACT

Educators agree that evaluation is an important part of the instructional decision-making process. However, evaluation of computer-related practice, projects, and policy at the local or regional level is not yet a regularly established component of the computer-use agenda in many educational systems. In this presentation, three individuals who are currently involved in the evaluation of computer-related activity at the district, regional, and international level will discuss strategies for program evaluation relating to regional computer activity and highlight benefits and contraints related to this level of evaluation.

When and Why Should Evaluation Occur?

Although not the case in theory, in practice the most common conception of evaluation is that of summative, judgmental activity. Frequently personnel involved at the district or regional level with the implementation of multifaceted computer activity comment that "it is too early" for evaluation, that "things are just getting started," and that teachers will be negatively affected by the thought of "being evaluated" when they are dealing with the complexities of managing technology in their classrooms.

In contrast to this perspective, evaluation can be seen as a formative procedure that is particularly appropriate when it is operating during the early phases of a particular project or when it is commissioned as a catalyst for district- or region-wide self-assessment that can enrich ongoing planning and policy making.

To illustrate these approaches to evaluation, two ongoing evaluation projects will be discussed. The first utilizes an external evaluator as a member of a project team for a nationwide educational software development project involving personnel from a Ministry of Education and a university as well as classroom teachers. The second illustration will describe the external evaluation of computer-related policy and practice in a school district where the evaluation is meant to provide insights into where the district should intensify its efforts and where it might "cut its losses" with respect to subsequent decision making involving computer-related technology. An important contribution that both these evaluations are making relates to the analyses of why things are not working in the way they were envisioned-sometimes this occurs because of "theory" problems (the original expectations were inappropriate or inappropriately expressed) and sometimes because of implementation problems. It is valuable for personnel at the management and policy level to be aware of the causes of system slowdowns in order to make informed decisions for subsequent planning or for immediate action.

How Can District-Level Process and Outcome Evaluation be Combined for Ongoing Activities?

Frequently both process and outcome evaluation are effective even while a districtwide project is only in the midstream phase. The next portion of the presentation will illustrate a methodology for such an evaluation, focusing on the Logo component of a districtwide computer education program at the elementary school level. The program, currently in its fourth year, underwent process evaluations each of the first three years. Evaluation of both process and outcomes are taking place this year although the program will be continuing. Various aspects of this system-level, internal evaluation will be described, including its purpose, design, and the design of the instruments used in the evaluation.

The presentation will also focus on constraints of the evaluation, concerns about the evaluation of outcomes, and difficulties encountered with the evaluation of outcomes. Finally, the ways in which the evaluation can help planning for the next year's evaluation will also be discussed.

The Problem of Evaluating Nonevents and Mismatches

in Computer Use

Before either formative or summative evaluation of computer impact on students at a district or regional level can take place, at least four major problems must be considered. Based on the evaluator's experience as an evaluator for an innovative mathematics curriculum implemented nationwide, the next portion of the presentation will focus on assessing problems in evaluating activity associated with changes in classroom practice. The four problems will be discussed relative to "nonevents" and "mismatches" between intentions and practice.

First, briefly stated, although computers may be available for student use, students' engaged time may amount to less than 10 minutes per student per week. Many evaluations report no engaged time with computers. Evaluation reports may describe the "effectiveness" or "ineffectiveness" of computer use when, in fact, computers have rarely been used. Second, when students or teachers use computers without an understanding of a program's operation and goals, the mismatch will affect what is learned. Third, even when program goals are clearly spelled out, a mismatch between students' levels of cognitive development and the cognitive tasks of the software may limit or preclude understanding. Finally, a mismatch between the skills acquired through computer use and the skills tested on standardized achievement tests may suggest that no learning has occurred when, in fact, significant progress in thinking or reasoning or writing has occurred.

In this presentation, ways to conduct evaluations to check for nonevents and mismatches with students and tasks, as well as match-ups with measurement devices will be discussed.

Conclusion

Despite the problems involved with conducting an evaluation at the district or regional level, the purpose of this presentation is to encourage such evaluations and to suggest workable methodologies. How to make evaluation a proactive and valuable contributor to ongoing policy and program development and implementation is the major thrust of the presentation.

MAKING COMPUTER EDUCATION FUN THROUGH APPLE COMPUTER CLUBS

Edith Kravitz

ABSTRACT

Apple Computer Clubs, funded by Apple Computer, Inc., is a program designed to improve educational computing by encouraging the use of microcomputers by people of all ages. This presentation will give an overview of membership benefits, ways to get a club started, the on-going support provided to members and how the program can be tailored to meet individual educational needs.

Apple Computer Clubs is designed to stimulate computer learning in a relaxed, educational environment. Students new to computing can pursue their interest in a non-threatening atmosphere and gain recognition for their progress. At the same time, more technically advanced students further develop their skills and enhance their leadership ability by sharing knowledge with their peers and advisors.

An excellent 10-minute video, produced by Apple Computer, Inc., will be shown during the presentation. This video explores how ACC advisors have successfully implemented the program with their students. Interviews with club members and their advisors and footage of activities such as the National Merit Competition, showcase the value of membership benefits.

Those attending this presentation also will have an opportunity to see projects created by club advisors and students. Ample time will be allowed for questions on any aspect of the program or presentation.

Developing a computer club is a wonderful way to channel youthful energy in positive directions. The Apple Computer Clubs Program seeks to make computer education interesting, stimulating and fun so that members can develop their talents and interests to the fullest extent.

COMPUTER LEARNING MONTH 1988
WHAT IS PLANNED FOR THIS YEAR & HOW CAN TEACHERS PARTICIPATE?

Sally Bowman
Director, Computer Learning Month
Palo Alto, CA 94306-6007

ABSTRACT

Technology is changing rapidly and teachers, parents and students need information to take full advantage of the benefits computers can offer them. Last October, with the help of state departments of education, national and local educational organizations, schools and educators, Computer Learning Month 1987 reached millions of teachers, parents and students with information on how to more effectively use computers for learning. Never before had there been such an fun, easy and exciting event to bring educators, parents, computer manufacturers, software publishers and retailers together for the benefit of children.

Computer Learning Month 1988 promises to be even more fun and exciting than 1987, as our primary goal is to expand the number of people receiving the benefits of technology in general. Computer Learning Month 1988's charter has been expanded beyond just learning with a computer in regular K-12 education and at home to also include colleges and universities, early learning, special education, career development and home uses of computers for learning, work and play.

Computer Learning Month has been endorsed by many national educational organizations who are working with us to help us achieve our goals. The list of endorsing organizations (honorary sponsors) is growing daily and includes National Education Association, American Federation of Teachers, National School Boards Association, International Council for Computer in Education, National Council of Teachers of Mathematics, American Association of Physics Teachers, National School Volunteer Program, National Foundation for the Improvement of Education, Association for Children and Adults with Learning Disabilities and Association of School Business

Officials International.

Computer Learning Month 1988 offers numerous ways for people to become involved individually and in groups, whether they know a little, alot or nothing at all about the use of computers. Computer Learning Month projects and books this year include:

- *Lesson Plan Books* for teachers in Grades K-6, 7-9 and 10-12
- Contests for students, teachers, community/school groups, schools & parents
- Career book: *Preparing for a Career in the 21st Century*
- Computer Learning Night/Day Kits for schools & community groups
- *University Software Resource Guide*
- *Family Activities Guide*
- *Everything You Need to Know About Computer Learning*

Computer Learning Month 1988 contest rules and entry forms and information on materials and local activities are available in the Computer Learning Month Newsletter that can be obtained by sending a postcard to the Computer Learning Month office.

INTEGRATION: ELEMENTARY/INTERACTIVE LASER DISC APPLICATIONS

Kim Becker, Monica Ellis, Kristen Hein, Ann Taylor
Kim Becker, Ann Taylor, Kristen Hein, Monica Ellis
Kitty Bateman, Helga Schwartz

ABSTRACT: Technology and the Future: An Elementary Approach

Kim Becker, Monica Ellis, Kristen Hein, and Ann Taylor, Indian Creek Elementary, Metropolitan School District of Lawrence Township, Indianapolis, IN

See below

ABSTRACT: Technology and the Future: An Intermediate Approach

Kim Becker, Ann Taylor, Kristen Hein, and Monica Ellis of Indian Creek Elementary School, 10833 East 56th Street, Indianapolis, IN 46236

Our society has been bombarded with massive expectations in the medical and engineering industries, computers and robotics have infested our businesses and homes, inventions and improvements in technological fields have skyrocketed. Meanwhile, society worries about the quality of the education their children receive and the effects of standardized testing on their future. Quality education including effective teacher/student/media interaction involving the sciences at an elementary level can address the above problem while preparing students early for a top quality future in our society.

Indian Creek Elementary of Indianapolis, Indiana was developed two years ago to accomplish that goal. It is a science/technology magnet school integrating science and technology throughout the curriculum including math, language arts, and social studies. Students utilize process skills and are actively participating with material such as robots, Lego-Logo, video disks, programmed software, word processing and varied printing skills, databases, graphics, and modems through AT&T.

Our presentation will outline just how science and technology can be effectively organized, maintained and taught at an elementary level. We will share our program in the form of a narrated slide presentation. Displays of student projects incorporating technology with various subjects will be available. Discussion and question-answer period is encouraged too.

The success of Indian Creek's program is evident through the visible products from children, their own enthusiasm for learning, and the creative thinking and process skills developed. We hope by sharing our unique curriculum that others can further their own students' education and enjoy teaching in a different mode.

ABSTRACT: A Collaborative Effort Using QUEST to Develop Interactive Mathematics Programs and Interactive Laserdisc Programs for Nonnative English Speakers

Professor Kitty Bateman, Department of Basic Skills, Queensborough Community College, and Dr. Helga Schwartz, Department of Mathematics, Queensborough Community College, CUNY, Bayside, NY 11364

As part of a Title III funded project, a faculty development seminar was instituted in which QUEST, a software package published by Allen Communications, is used as a means of encouraging faculty to develop CAI for the disadvantaged technical students attending the college.

The mathematics component of this project was written for technology students taking a trigonometry and algebra course. One of the most difficult subjects for these students is trigonometry in the coordinate system. This software is an interactive tutorial that introduces the students to the naming of the angles in the coordinate system; the definitions of the three trig functions--sin, cos, and tan; and gives the students practice in finding angles and their trig functions. The graphics capabilities of QUEST are significant and are demonstrated in this program.

A second part of this project involved the collaboration of a mathematics professor and an English as a second language faculty member in the creation of an interactive lesson using a videodisc, laser disc player, and an IBM PC. As a consequence of the specific features of laserdiscs, that is, their ability to present to nonnative English speakers examples of natural comprehensible speech, the lesson focuses on the following skills: listening for information, recalling details, developing vocabulary, making assumptions and drawing conclusions. In addition, there will be some attention to the collaborative model and its implications for the creation of CAI within the CUNY system.

POTPOURRI

Gray Prewitt
Donald Pratt

ABSTRACT: Computer Literacy for Administrators

Gray Prewitt, Mississippi Educational Computing Assn., Itawamba Community College, Tupelo, MS

Administrators in Mississippi have experienced a steady increase in the volume of information which must be managed by their offices. While the solution to use computers to ease the burden was obvious, many found the task of selecting equipment and software, as well as using it, not at all easy.

The Mississippi State Department of Education, realizing the need for assistance, is providing through their Mississippi School Executive Management Institute a training module designed to provide administrators with the basics needed to begin making these decisions and using this technology.

The "Computer Literacy for Administrators" module consists of five units:

- Computer Literacy
- Word Processing
- Database Management
- Spreadsheets
- Planning

and is presented in a five day workshop managed by two trainers. The materials for the workshop include a manual which covers all units of the module, video tapes and slides which may be used as enhancement and a complete set of software which the administrator receives upon completion of the training. The workshops are offered in various locations across the state several times during the year both during the school year as well as in the summer. With instruction designed to maximize hands-on experience, the Community Colleges and Vocational Centers have proven to be the ideal locations since they are able to provide sufficient equipment for a one to one ratio.

Evaluations conducted on a daily basis during each of the workshops have indicated widespread acceptance and approval by all the administrators who have attended. The participants come to the workshops with enthusiasm and leave with new skills which will be useful in the performance of their jobs.

ABSTRACT: Teaching SuperPILOT with SuperLESSON

Donald L. Pratt, Department of Curriculum and Foundations, Bloomsburg University, Bloomsburg, PA

Learning SuperPILOT is at once a rewarding and frustrating experience. It is a very simple language to use once learned but is quite frustrating until one gets an overall picture of how the pieces fit together. SuperLESSON, a noncommercial SuperPILOT tutorial developed by the author, gives students this overview in a friendly setting.

The program usually takes students about 45 minutes to complete. Students are quite impatient about learning what they want to know, and it has been observed that they readily take advantage of the menu-driven multiple branching features of the table of contents and index. They have indicated that SuperLESSON has helped them understand relationships that had eluded them.

The table of contents of SuperLESSON is subject related (e.g., graphics, sound), and the index is command related (e.g., as:, ts:). This allows students to select the topics of their choice or to use the program as a reference tool. Text entry, graphics, and sound are introduced first so as to get users quickly and simply into actual lesson construction right away. Simple screen design hints are suggested throughout, and the more difficult subject of answer handling techniques is introduced last.

The author has conducted a preliminary effectiveness study of SuperLESSON which shows that some students are helped when SuperLESSON is used to first introduce SuperPILOT's structure. Not many used it as a reference tool. An instrument was developed to assess student opinion about its effectiveness. The instrument was used before and after SuperLESSON was used in the SuperPILOT section of the course. One class was introduced via SuperLESSON, and another was taught without using it.

Much patience is required in teaching SuperPILOT, but persistence pays off. Students are very creative and derive a great deal of personal satisfaction in making their own computer materials, a feat few teachers can boast of. Giving students an early overview of how SuperPILOT works and providing much moral support are the two main ingredients essential to students staying with their projects until they master them. SuperLESSON can be a useful tool in the early, critical stages when apprehension and discouragement are the highest.

PRE-COLLEGE POTPOURRI

Karen Hoelscher
Antonia Stone
Karin Wiburg, Maria Wilson

ABSTRACT: Using Formative Evaluation to Improve the Design of Interactive Videodiscs

Karen J. Hoelscher, Harvard Graduate School of Education, Cambridge, MA

In the past decade, an increasing number of teachers and students have used microcomputers as part of the school curriculum. However, in spite of increased use of educational software, many question its effectiveness (Becker, 1987; Chen & Paisley, 1985; Pea & Sheingold, 1987). A substantial number of software products have been published; some are considered more effective, many are not. As a result, educators and researchers have called for more systematic methods of collecting and implementing feedback from teachers and students at specific stages of the software development process in order to improve the quality of the final product (Flagg, in press; Gagne, 1987; Weston, 1987).

There is a limited amount of published research about this topic--a process called "formative evaluation"--for several reasons. Among the issues most often discussed are: (a) The field of educational technology is relatively new; (b) formative evaluation of software is an expensive process which is often conducted quickly with a small sample, and therefore not usually considered "scientific," and (c) results of these evaluations are sometimes considered confidential. A review of the literature has outlined the questions and problems this type of evaluation raises for developers of educational software.

In view of these issues, the author has addressed two questions in a review of the *unpublished practice* of formative evaluation. First, how do software developers make decisions about, conduct and implement formative evaluation during different phases of the development process? And second, how do characteristics of a particular software application (e.g., evaluating a series of products versus a single product or evaluating a tool program versus a drill and practice program) affect these choices? Knowledge of how evaluation plans differ for various types of software--and how and when those plans are used--are the basis of the author's development and testing of various models for collecting and using feedback from teachers and students to improve the design of software products.

The presentation will contain two parts:

1. An overview of the literature and practice of formative evaluation of educational software, based primarily on interviews with practicing software developers across the United States; and

2. A preliminary report on the formative evaluation of an interactive videodisc project codeveloped by Harvard Law School and the Rochester (NY) Institute of Technology. The videodisc provides a *workstation* for law school students to practice building cases. During its development, feedback from students, law professors, and practicing attorneys is being used to improve the final design of the disc.

ABSTRACT: Templates for Literacy

Antonia Stone, Project Director and Executive Director, Playing To Win, New York, NY

Although the need for literacy learning among adults has been the subject of increasing attention in the media, and while millions have been spent to ensure that children in school have access to computer tools and learn how to use them, little has yet been done to support the need of the adult nonreader or very beginning reader for technological awareness and "information literacy."

Playing To Win, a nonprofit dedicated to improving opportunities for access by disadvantaged people to computer technology and education, is engaged in a three-year project funded by the NYNEX Foundation to document its successful approach to using the computer with adult nonreaders. The project will produce a book of guidelines and strategies for incorporating word processing, spreadsheet, database, and graphics software into literacy curricula.

This presentation will include:

1. Demonstration of sample project material suitable for use by small classes or one-on-one tutoring programs;
2. A report on national pilot testing of project materials in three diverse literacy programs during the spring of 1988; and
3. Step-by-step guidance for producing original

templates.

The project materials will be of interest to those concerned with the provision of literacy instruction to adults and also to those seeking creative applications of tool software for learning at any age.

ABSTRACT: Impact of Computer-Based Learning on Elementary Students Cognitive Abilities: A University-School District Collaborative Research Project

Dr. Karin Wiburg, Assistant Professor, Computer Education, United States International University
Dr. Maria Wilson, Computer Specialist, Park Dale Lane Elementary, Encinitas School District

THE PROBLEM

The purpose of this university-school district collaborative study was to compare the effect of different types of computer use on the cognitive abilities of fourth grade students.

METHOD

Sixty-nine fourth grade students at Park Dale Lane Elementary School in Encinitas, California participated in this study. One experimental group used programming activities to create projects based on the content of the regular fourth grade curriculum. A second experimental group used applications software (word processing and a graphics package) for learning activities based on the same content material. An instructional model for computer-enhanced problem solving was developed and used for both experimental groups. The model involved 1) teaching programming or applications within a knowledge-based context and 2) teaching problem-solving strategies. Students in a third control group received no special instruction, but continued to participate in the regular computer education program, using primarily computer-assisted instructional software (C.A.I.). The treatment groups each spent fourteen forty-five minute sessions in the computer lab over a four month period. During this time the control group continued to attend the computer lab on the same regular basis as all students in the school, which constituted approximately eight forty-five minute sessions.

Both process and product evaluation were utilized in the study. Students were pre and post-tested using the Developing Cognitive Abilities Test (DCAT). There were no statistical differences between the groups prior to the study in terms of higher-level cognitive abilities, measured in this test as the "Application, Analysis, and Synthesis" levels of Bloom's Taxonomy. Both of the treatment groups produced written products using the computer, which were evaluated by English teachers, using an instrument based on the 1987 California Assessment Program's (CAP) Direct Writing Assessment.

RESULTS

At the end of the experimental study, both the programming and word processing groups scored significantly higher in higher-level cognitive skills, as measured on the DCAT., than the control group. While the programming group made the highest gains, they did not differ significantly from the word processing group on this test.

Products produced by the programming group using LogoWriter were rated significantly higher than those produced by the word processing group, particularly in the area of divergent production, a measure of creativity. The research supports the need for further investigation of the power of student-controlled word processing and programming environments to enhance student learning, particularly in terms of their positive impact on higher-level cognitive abilities.

UNIVERSITY MICROCOMPUTER APPLICATIONS: HEALTH/MEDICINE

Judith A. Barlow, Joseph S. Morrell, Cheryl J. Norton
JoAnne Sparks
John A. Ullian, Michael L. Noel
Suzanne Brisk, Ernesto Randolfi

ABSTRACT: ISWELL: An Expert System for Wellness Assessment and Exercise Prescription

Judith A. Barlow, Assistant Professor of Computer and Management Science,
Joseph S. Morrell, Professor and Chair, Computer and Management Science,
Cheryl J. Norton, Associate Professor of Physical Education, Recreation and Health, Metropolitan State College, Denver, CO

In spite of interest in health and fitness, effective wellness assessment and individualized exercise prescription are not readily available to most people. We believe that this is due in part to the time consuming processes of manual record keeping and data analysis, as well as a shortage of qualified experts for exercise prescription. These services can be made available to a wider range of individuals by using an expert system to assist in the wellness assessment/exercise prescription process.

ISWELL, the Intelligent System for WELLness, is an expert system for use in wellness assessment and exercise prescription. Client data, including relevant medical history and personal information, are entered directly into the computer. Data related to measures of cardiovascular functions, body composition, pulmonary functions, strength, flexibility and endurance are compared with corresponding values in the normative database and used to generate the individualized wellness profile.

A knowledge base of logical facts and rules regarding health, fitness and the effects and interactions of various exercises is created by applying artificial intelligence methods. This knowledge base interacts dynamically with the database to evaluate an individual's level of wellness and interactively with the client or fitness advisor to develop an appropriate exercise plan.

Client data are entered manually at this time; however, future plans include incorporating a means of inputting client data on-line directly from the measuring instruments (e.g., electrocardiograph, sprirometer, sphygmomanometer, skinfold calipers, etc.). The client's scores in the wellness areas are compared with corresponding values in the normative database and used to generate the individualized wellness profile.

Once the client's current situation has been evaluated, the client's health goals are considered. Health goals might include improvement in a specific wellness area, conditioning for a sport or activity, overcoming or compensating for specific health problems, or preventing anticipated problems. After comparing the requirements for meeting the health goals with the current measures in each of the wellness assessment areas, an individualized exercise plan which leads to improvement in the desired areas is developed.

ISWELL will help the client and/or fitness advisor make more intelligent decisions in choosing the most appropriate exercise plan and can also be used to teach fundamental concepts of exercise prescription.

ABSTRACT: The Development and Implementation of a Medline Database Searching Course for Third Year Medical Students Using Programmed Instructional Methods in a Laboratory Setting

JoAnne Sparks, Reference Manager, Library of Rush University, Rush-St. Luke's-Presbyterian Medical Center, Chicago, IL

The primary objective of this project is to teach third year medical students how to design and execute a search strategy on the Medline database using a microcomputer and telecommunications software. Later phases of the project will include other groups of clients in the medical center including residents, faculty, and researchers.

An workbook was developed using a programmed instructional approach. The manual contains real clinical case studies and takes the student through the logical process of performing the search on the database. System commands and telecommunications commands are provided on quick reference guides and/or verbally as necessary. There is no emphasis on memorizing commands. Rather the emphasis is only on learning the thought process.

The final course configuration evolved into the following format: Students receive course materials at the beginning of their clerkship. Each group is already divided into teams of three. Each ward team sets up an appointment with one of the course instructors. The manual requires approximately 45 minutes to 1 1/2 hours to complete. Each group is encouraged to work at their own pace. The final exercise is to construct a real search strategy for one of their own clinical questions which will be used in the patient care setting.

The course is designed to be flexible in order to accommodate students' schedules and to focus on useful, practical information that is relevant to their current and future informational needs.

Outcomes at this point are primarily positive. The course has developed into an extremely flexible learning experience and can be adjusted for schedules and level of expertise of the student. The information retrieved during the course and the skills learned are immediately applicable to the research and patient care activities of the students.

ABSTRACT: Using Computerized Patient Management Problems in a Medical School Classroom

John A. Ullian, Research Associate
Michael L. Noel, M.D., Assistant Professor,
Department of Family Medicine, Baylor College of Medicine, Houston, TX

Computerized patient simulations, or patient management problems (PMPs), require that the user make decisions typically made by physicians in a patient visit, including what to ask in a medical history, which aspects of a physical exam to perform, and which tests to order. This requested information is provided, and the user makes a diagnosis and prescribes treatment. Primarily produced for continuing education of physicians, they are also available for use by medical students.

We use commercially available PMPs from several sources in the classroom with second year medical students. This is unusual in that the PMPs were created for individual, not classroom, use. The class is "Introduction to Clinical Specialties," a required course with 162 second year students. It occurs at the transition between basic science courses and clinical rotations, and its objectives include the teaching of clinical problem solving methods. Since scheduling 162 students to use PMPs individually to practice problem solving is difficult logistically, we decided to use PMPs in class.

We run the PMPs on a Compaq Portable I, using a Kodak Datashow projection pad to project the monitor's image onto the screen. At each decision point (e.g., select history questions to ask, select a part of the body to examine, enter differential and final diagnoses), students are asked to make their own decisions. The instructor then enters the appropriate decisions on the computer, discussing the reasoning behind his choices. Alternatively, students have been polled and the most popular decisions entered.

While we consider individual use of PMPs valuable, and encourage our students to do so to supplement the classroom presentation, we have also found classroom use of the PMPs valuable. The PMP provides a clinically meaningful structure for a presentation. It allows "mini-lectures" on topics not well understood by the class, as well as correcting or updating material presented by the program. It has encouraged additional student-teacher interaction in this course. It allows the student to practice clinical problem-solving without potentially harming patients. It models the systematic data gathering, hypothesis generating, and hypothesis testing methods so necessary to clinical problem solving. Finally, it introduces students to the PMP as an educational tool they can use now and in their careers.

ABSTRACT: Comparative Review of Health Risk Appraisal Software

Suzanne Brisk, BS, and Ernesto A. Randolfi, PhD,
Department of Health and Physical Education, Syracuse University, Syracuse, NY 13244

A computerized health risk appraisal (HRA) is an educational tool used to predict the risk of developing illness or early death. Students answers to questions regarding lifestyle and health-related behaviors are entered into the computer and then compared to known risks of morbidity and mortality. Each student is then given an individualized assessment of his or her relative health risks with suggestions of how to decrease health risks. Most programs also offer an aggregate evaluation for groups of students. Educators can use this summary data to target risk factors and plan for health education programs.

The need for this study was encouraged by the increasing use of HRAs as educational and motivational tools. According to a recent government publication, there are 50 HRAs available consisting of self-scored questionnaires, microcomputer-based appraisals designed for batch processing on mainframe computers (Healthfinder, Office of Disease Prevention and Health Promotion, 1986). The availability of this many programs designed for the same purpose has created confusion for health educators in the market for an HRA.

Thirty-four companies that publish microcomputer-based HRAs were asked to participate in this study, and 13 agreed to the review. Appraisals that were targeted for specific populations or specific health concerns (e.g., cancer risk) were not used. A total of 10 HRAs were evaluated on the basis of their educational utility.

The software evaluation process included an assessment of price, preview policy, backup policy, hardware needs, update provisions, scoring, clinical measures, reliability, educational quality of individual printouts, source of database for the normative population, educational utility of compiled reports, package documentation, and guidelines for consumer and educational use. For each software package a sample analysis was prepared of a standard male and female health profile.

A comparative analysis of the software and output of the sample analyses demonstrated that HRA software packages vary considerably in the quality of their specific features. A summative ranking of the 10 HRAs was not attempted. Individual health educators will have to assess which evaluation criteria are most important to them and which package best meets their needs.

This investigation allows health educators to see the range of differences and similarities in microcomputer-based HRAs. Additionally, it offers a comprehensive set of suggested criteria that health educators can use to make informed choices in software selection. Careful software selection may in turn keep the demand for and development of quality HRA programs at a high level.

ADVENTURES IN MATHLAND: AN INTERDISCIPLINARY APPROACH TO K-6 MATHEMATICS INTEGRATING COMPUTERS, VIDEODISC, AND MANIPULATIVES

Ted M. Kahn
Marvin M. Zauderer
Ann H. McCormick
Picodyne Corporation

ABSTRACT

Adventures in Mathland is a comprehensive, multimedia project designed to meet the needs of K-6 mathematics education. Based on the new California Mathematics Framework, this series of products provides interactive, challenging, discovery learning experiences through a blend of microcomputer software, manipulatives, offline activities, and the Mathland Museum videodisc. Mathland's activities are linked to ancient and modern cultures, and help children to make deep connections between mathematical ideas and world history. The project is one of those supported by the Software Development Partnerships program of the California State Department of Education.

This presentation will present some of the rationale, background, and action research related to the development of these multimedia products. In addition to discussing the design considerations, we will also present data from a unique group of pilot schools, all of which are linked by an experimental "field test telecommunications network."

VARIATION IN AN INNOVATION USAGE PATTERNS FOR THE VOYAGE OF THE MIMI

Regan McCarthy
Math, Science and Technology Teacher Education Project
(MASTTE Project)
Bank Street College of Education
New York, NY

ABSTRACT

In recent years, much attention has been given to "change" in schools. For more than a decade, researchers from Berman and McLaughlin (1978) to Hall and Hord (1987) have emphasized, warned, and cajoled us to view interventions in schools as a process involving personal and structural change. Further, many researchers concerned with change go so far as to say that attempts to introduce innovations in schools without addressing change issues are doomed to mediocrity - or worse - failure. But how is change manifested in instruction in a given classroom over time? What does change mean to a teacher asked to do new things in teaching, "things" which may involve completely new content, pedagogy, and technologies?

Some interesting answers have been provided through the MASTTE Project, a national demonstration, research, and dissemination effort funded by NSF to help teachers and staff developers improve math and science instruction and the uses of technology in math and science. The MASTTE Project has worked with teachers and staff developers from nearly two dozen school systems in nine areas of the country. The Project has four broad goals: to help participants improve their understanding of science and math content and concepts; to gain a clearer grasp of a sound pedagogy and philosophy of science and math instruction; to assist teachers in the planning and integration of science and mathematics instruction in their schools and classrooms, and to assist staff developers to plan and implement a program of training and support in these areas; and to help participants use new technologies appropriately in science and mathematics instruction.

These goals are achieved in part by use of a powerful and engaging set of materials--The Voyage of the Mimi--which acts as a catalyst for work related to the Project's goals. These materials integrate three media into a cohesive set of supplementary materials which introduce and reinforce science and math concepts: a television series combining an adventure

drama and documentaries; microcomputer software; and print materials. Software and print materials are organized into four learning modules that are based upon content presented in the television series.

Since our aim is not specifically to train participants to use Mimi materials, but rather to use Mimi as a means to a considerably broader end, we have presented many ways to use Mimi materials to teachers and staff developers. Consequently, an important part of the Project's research has been to identify the ways in which Mimi teachers use these materials initially, how use changes over time, what factors seem to influence the ways in which materials are used, and how effective different uses of the materials are.

We have identified four distinct patterns of use of Mimi materials: video-only; module-based; discipline-based, and theme based. In this session we will describe each approach in detail and will show and use videotape segments of teachers using each approach. Subsequently, we'll discuss the advantages and disadvantages of each approach, the change observed in teachers over time, and preliminary findings on effectiveness. The implications of our observations for others helping teachers to use integrated technologies in math and science instruction will also be discussed.

PRE-COLLEGE INSTRUCTIONAL TECHNOLOGY DELIVERY SYSTEMS

Bill Ashby
Robert Caldwell
Leroy J. Tuscher, Pete Geyer
Mary Furlong

ABSTRACT: Having a Brainstorming session? Take a CAB (Computer Assisted Brainstorming)

Bill Ashby, Consultant for Computer Assisted Instruction, Arrowhead Area Education Agency, Box 1399, Fort Dodge, IA 50501

CAB is based on a concept suggested by E.M. Stacy of the Iowa School for the Deaf, Council Bluffs, Iowa, developed and enhanced by Bill Ashby, Arrowhead Area Education Agency, Fort Dodge, Iowa. Both have used it in a variety of settings for work with groups.

Brainstorming has gained wide acceptance as a group technique for problem solving. The use of the computer, a large screen display and a database can assist in the capturing of ideas as they flow, the organization of the ideas into meaningful groupings, development of a sequence of ideas and creation of final product.

A database is created with appropriate fields for inputting the information during the brainstorming. The database is displayed through the use of the largescreen display connected to the computer to provide interaction of the entire group. As ideas are suggested by group members the recorder enters them in the database. Ideas can be given and entered on the computer in any order . The visual display allows the entire group to be involved in the brainstorming session.

When the collection of ideas is completed the ideas that were presented can be reorganized according to any field that can be sequenced. The use of the data base and the visual display allows the entire group to be involved in the process of sequencing the information. The use of the database allows for rearrangement of the information and the adjustment of concepts that were presented.

If integrated software, such as Appleworks, is used the information in the database can be moved to the wordprocessor for more refinement of the model and producing a final product from the information gathered during the session.

This presentation will present a demonstration of CAB and how it can be useful for the organization of a group information gathering activity. The same CAB process can be useful for recording any type of meeting where information is being discussed and a report of the information is needed. The computer can be used to organize the discussion process and to refine the final report from the group. Today's hardware makes the microcomputer a powerful to aid the process of creative thinking.

ABSTRACT: Can We Talk? An Audio Record and Playback System for the Apple IIGS for Creating Speech-Supported Lesson

Robert M. Caldwell, PhD, Vice President, Education Division, Learning Technologies, Inc., Dallas, TX

Recent advances in integrated circuit design, graphics, and voiced and sound production have increased the capabilities of personal computers to maximize their usefulness and potential as teaching tools. Advanced computer systems such as the Apple IIGS have dramatically increased the ability of computers to introduce auditory as well as visual and kinesthetic stimuli into the computer learning environment and to offer a broad range of possiblities for providing instruciton to children of all ages and particularly to certain student populations who formally were restricted and even excluded from using computers.

Although few educators would deny the enormous potentional of using computers to create learning environments where voice, sound, graphics and text are successfully integrated, commercial applications of important features such as voice and sound technology have been slow to develop. Industry analysts have cited a number of reasons why publishers and hardware manufacturers have been reluctant to incorporate voice and sound technology into their products. The following are two of the major reasons:

1. Quality of Voice and Sound Reproduction

To date, voice and sound generated by a computer has been produced by one of two techniques: voice synthesis and voice/sound reproduction. For the most part, most all efforts to synthesize the human voice have ended up sounding mechanical and , in some cases, unitelligible. As a result, inaudible or indistinguishable speech has had limited value as a learning aid for most student populations.

Voice and sound reproduction, on the other hand, involves a process for recording specific words or sounds and reproducing them later without the aid of add-on equipment. The result is highly recognizable speech, but its range is restricted to only those words or sounds that have been prerecorded on a disk, usually in a studio using expensive recording equipment.

2. Additional Hardware

A disadvantage common to both voice synthesis and voice reproduction is that they both require addtional hardware to produce voice and sound as input or output. Use of voice synthesis usually requires a piece of additional hardware to "speak" words or create sounds. With voice reproduction, it is possible to reproduce voice and sound without the aid of additional equipment; however, to record voice and sound, users must have the proper hardware to create and store the vocabulary. In most cases, this equipment is not readily available to the average educator, parent or school district.

The Audio Record and Playback System (ARPS) takes advantage of the features of a voice reproduction technique called voice-store-and-forward. This system, when fully functional would allow parents and teachers to create lessons which include high quality speech and audio stimuli and reinforcements without expensive add-on equipment. This presentation demonstrates the capabilities of the ARPS prototype which is designed to take advantage of the sound capabilities of the Apple IIGS and focuses on the potential applications of the system for populations such as beginning readers, ESL and bilingual education, adults and learners who have special needs.

ABSTRACT: Small and Large Group Applications of Interactive Videodisc

Dr. Leroy J. Tuscher, Director, Educational Technology Center, Lehigh University
Pete Geyer, Research Associate, Educational Technology Center, Lehigh University, Bethlehem, PA 18015

Educational technologists have been exploring interactive video for more than 10 years. In most cases, the emphasis has been on individualized applications; that is, the technology teaches and the student learns. We argue that more consideration should be given to group applications, especially when the targeted environment is the regular elementary or secondary classroom.

Small group applications involve several students at once in a learning activity. For example, students studying the artists Vincent Van Gogh and Paul Gauguin could have access to a videodisc-based art visual database. The students would interact with each other to determine appropriate key words for searching the database for relevant visuals. They might decide to narrow their search to "only those paintings done by Van Gogh or Gauguin in the city of Arles." The students interact with the technology, but more importantly, they interact with each other.

Large group applications engage an entire class of students at once. A teacher or student leads a discussion on a topic, calling up relevant visuals (computer and video generated) as needed. For example, a teacher presenting a unit on whales might involve the class in a discussion about behavior of whales. A student asks "What does breaching mean?" The teacher responds, "Let's take a look at breaching in action", then uses a remote control device to select "breaching" from a visual glossary. In an instant the students' eyes are treated to the sight of a whale leaping out of the water and splashing down on its side. The teacher assumes a familiar and desired role, and a single hardware configuration is used effectively with an entire class.

Group applications of interactive videodisc are especially realistic given that schools don't have many videodisc players. If we can develop such applications for teachers, then we can begin to seriously consider integrating the technology into the curriculum.

ABSTRACT: The Use of Hypermedia and HyperCard™ in Education

Mary Furlong, Ed.D., Betsy Franklin, Janet Graeber, Roger Knight, Lenny Schartz

This University of San Francisco recognizes the high educational value and practical application that hypermedia now has and will have in the future for the widely varied uses of computers in education. The University of San Francisco is actively exploring immediately available methods for using HyperCard™ a hypermedia authoring tool, to develop and present curricula for kindergarten through the 12th grade and for higher education. The methods currently being explored include: multimedia (control of text, audio, and visuals), computer assisted instruction design and implementation, student and teacher computer literacy, and computer programming instruction. Graduate and undergraduate students at the University of San Francisco have been excited about and developing educational applications for HyperCard™ both as requirement for university courses and on an independent basis since the introduction by Apple Computer of this innovative product on August 1, 1987.

INFUSING THE COMPUTER INTO THE CURRICULUM: A WORKSHOP APPROACH

Edward L. Vockell
Purdue University Calumet
Hammond, Indiana

Helen Jancich
Crown Point School System
Crown Point, Indiana

This paper describes how a Teacher Incentive Grant was used to provide a series of workshops to teach 40 teachers to use the computer effectively. The training program was offered not with "technology" funds, but rather as part of a Teacher Incentive Program. The stated purpose of the program was to help teachers (1) develop a familiarity with the basic applications of microcomputers, and (2) use the computer to teach and to extend higher order thinking skills.

The training project was enthusiastically received by the participants. They viewed it as a genuine "perk" to have the opportunity to learn about microcomputers. The project clearly met its objectives. This report summarizes the main features of this training program, provides data to evaluate the program, and offers suggestions to enable other school systems to offer similar training programs. In addition, this report suggests activities that can be offered as a follow-up to this project. Finally, this report describes specific types of software teachers are likely to use both with and without a training program to introduce them to instructional applications of the microcomputer.

Infusing the Computer into the Curriculum:
A Workshop Approach

In the fall of 1986, the Crown Point, Indiana, school system had substantial amounts of microcomputer hardware and software, but these materials were not being used as effectively as possible by many of the teachers. This paper describes how a Teacher Incentive Grant was used to provide a series of workshops to teach 40 teachers to use the computer effectively. It is important to note that the training program was offered not with "technology" funds, but rather as part of a Teacher Incentive Program. The stated purpose of the program was to help teachers (1) develop a familiarity with the basic applications of microcomputers, and (2) use the computer to teach and to extend higher order thinking skills. The participating teachers definitely viewed the training program as a genuine "perk."

This paper presents two useful kinds of information: (1) it describes the main features of a training program to encourage teachers to use computers and evaluates that training program, and (2) it describes the activities typically carried out by teachers who have access to computers in a large suburban school district, both before and after a training program.

Project Activities

Workshop Sessions: There were two major (6-week) series of workshops on Saturday mornings. The first series was open to elementary and middle school teachers, and the second to middle and secondary school teachers. In addition, there was a pair of follow-up workshops at the very end of the program. Each Saturday workshop session lasted four hours. Therefore, each participant attended seven four-hour sessions. Participants was paid a stipend of $35 to attend each Saturday session. In addition they were eligible for one graduate credit from Purdue University Calumet by completing appropriate requirements.

The major thrust of the 6-week workshops was to enable teachers to use computers to teach higher order thinking skills. However, in order to accomplish this, it was necessary to permit the participants to explore the application of the computer in areas of immediate interest to themselves which may not be directly applicable to the teaching or reinforcement of thinking skills. Such exploration was encouraged. Our experience has been that teachers will use the computer effectively only if they already perceive it as a helpful tool. For this reason, we demonstrated and and encouraged participants to use gradebooks, word processing programs, databases, drills, tutorials, and simulations which were of immediate interest as well as programs more directly related to thinking and problem solving.

Individual Pre-Workshop Consultations with Participants: After acceptance into the program and prior to the workshops, each participant had an individual conference with one of the directors of the program. The purpose of this consultation was to enable the directors to gear the workshops as specifically as possible to the needs of the participants. As a result of this conference, the directors had a specific plan regarding computerized materials for each teacher to work with during the workshop sessions.

Individual Post-Workshop Consultations with Participants: After the 6-week workshops and before the follow-up session, each participant had a second individual conference with one of the directors. The purpose of this second consultation was to help the teachers transfer the information and techniques from the workshop into actual classroom practice.

Content Modules: In addition to the 6-week workshops and personal conferences, supplementary content modules were available, which focused on topics specified by the teachers and selected after their pre-workshop conferences. Participants could attend up to four of these half-day workshops, which were taught by the project directors and by two additional instructors with expertise in designated areas.

Materials: The main materials were a set of about 100 programs on loan from various companies and the MECC materials which had been in the Crown Point system but had not been actively used by many teachers. Most of the programs related to thinking skills in general (e.g., logic or memory skills, concept development, etc.) or to higher order skills within a subject area (e.g., testing hypotheses in science, using the subjunctive in French, etc.). Additional programs (e.g., many of the MECC programs, Print Shop, and Crossword Magic) were programs that encouraged the use of the computer but did not focus on higher order skills.

Each teacher received a supply of 50 disks on which to make copies of software which they wished to use in their own classrooms. These disks have remained the property of the individual teachers, increasing the likelihood that they will use them in their own teaching activities. Experience shows that the combination of easy availability of software and familiarity with its use will lead to effective implementation of the materials to reach curriculum goals.

Teachers are often frustrated and avoid effective computer software, because they would have to alter their entire curriculum to incorporate the new ideas. A major goal of the proposed program was to help teachers identify useful programs to meet existing, important curriculum goals.

The project's combination of voluntary participation, token stipends, graduate credit, individual conferences, training sessions, content modules, and follow-up activities was designed to make the teachers want to learn as much as possible about using the computer to teach high order thinking skills. The project deliberately incorporated psychological principles of motivation, organizational change, and adult learning to make it possible for teachers to view computers as a very useful tool to solve their problems and bring them fulfillment, instead of as another burden with which they must deal.

Program Evaluation

Table 1 summarizes the results of a questionnaire the participants completed after the post-workshop conference. Since the questionnaire was anonymous, it was not always possible to determine who had not returned a questionnaire, and hence only 32 of the 40 participants responded. These results show almost unanimous satisfaction with the project. In general, the participants enthusiastically agreed that the computer facilities were good, that adequate software was available, that stipends and length of sessions was appropriate, that the pre and post planning conferences were useful, that the instructors were competent and helpful, that they would be able to apply what they had learned, that they would like to take a follow-up series of workshops, that there was a good mixture of practical and theoretical information, and that the optional three graduate credit hours was a good idea. On the other hand, while no one strenuously opposed the one hour of graduate credit, only 41% indicated that it was a strong incentive to participate. In addition, only 9% said they would have refused to participate without the stipend, whereas 50% explicitly denied this. Only 82% stated that their supervisors or superiors had been supportive.

Insert Table 1 about here.

The impact of the workshops is reflected in the number of participants who stated in the post-conferences that they either began or accelerated their use of computers in their classrooms. Of the 40 participants, 52 percent indicated that they had essentially began to use computers for instruction as a result of the workshops. Of these, over half had already become avid users by the time of the post-interview, and most of the others showed promise of using the computer much more often in the coming year. Another 35% indicated that the workshops had led to an acceleration in computer use. The five non-accelerators indicated that they had already been using the computer about as much as it could be used and therefore could not honestly say they had accelerated their use of the computer. However, even these five indicated that they were satisfied that their selection of programs and the effectiveness of their use of the computer had improved as a result of the workshops. In addition, most of the teachers stated that they had found useful ways to use computers to stimulate higher order thinking skills.

One of the major outcomes of the project was that teachers became familiar with programs already available (but relatively unused) within the school system. Nearly all the teachers indicated that they appreciated the time to sit down at a computer and to examine software without pressures to complete other tasks. Crown Point has belonged to MECC for four years, but the software has not been heavily used. During the project, each teacher received 50 floppy disks for their own use. As a result of viewing programs during the workshops, the 40 participants requested copies of approximately 2000 MECC programs, which they stored in suitable containers for easy use in their classrooms. During the post-workshop interviews, this easy availability of the MECC software and familiarity with its contents were frequently cited as valuable outcomes of the project.

A valuable outcome was that participants learned new strategies for using the computer in their classrooms. For example, most of the teachers had previously permitted students to use the computers either singly or in very small groups (usually 2 students). During the workshops, the instructors demonstrated the use of a single computer attached to a large screen as a focus for a whole-class activity.

Suggestions for Improvement of the Workshops

When asked how to improve the workshops, nearly all the respondents stated that the workshops were fine the way they had been run. When pushed to give suggestions for improvement, however, they supplied these ideas:

1. Get software from a wider variety of companies.
2. Give more lead time and a better description of the workshops.
3. Enlist more enthusiastic support from building principals and department heads.
4. Be sure the content modules are clear and practical.

Suggestions for Follow-Up Activities

When asked what type of follow-up activities would be useful, the participants gave numerous suggestions. The following is a list of those most likely to be useful:

1. Provide more previewing sessions similar to those conducted during this project.
2. Provide training sessions devoted to specific topics.
3. Provide guidance for teachers working on specific projects.
4. Provide supervised microteaching.

5. Provide workshops for people with common curriculum interests (for example, third grade teachers).
6. Provide for the administrators a series of workshops to enable them to become familiar with instructional applications of the computer.
7. Several participants stated that they could provide more useful suggestions after they have had time to let the results of this project "sink in."

In general, these suggestions point to a need for continued updating to enable teachers to keep their level of knowledge current and for more detailed information regarding specific and specialized pieces of courseware. Many of these suggestions could be incorporated into regular inservice training opportunities.

Computer Use Survey

Before the training program, in April 1986, a survey was conducted to determine how often teachers used the microcomputers and what software they most often used. A similar survey was conducted in October 1987, a semester after the completion of the training program. A question was added to the followup survey to identify which teachers participated in the training program.

Table 2 summarizes the major results of this survey. The most obvious result of the training program was an increased number of teachers who said they used the computer (100% among participants). These increases in computer usage came about through increased use of existing facilities; there was not a large increase in computer hardware between the two surveys. Not only did the workshop participants show considerable overall increases in the use of the computer; but they also began to use the computer for higher order activities (e.g., simulations) more often than either of the other two groups.

Insert Table 2 about here.

Table 3 summarizes the "top ten" software choices for each group of teachers. This table shows not only an increased usage of some of the familiar software packages, but also an increased use of tool programs (such as AppleWorks, which can be used as a tool in many subject areas) and of higher order programs (such as the Tom Snyder materials and Rocky's Boots). Almost all the software reported by teachers on the posttest was also available (but relatively unused) at the time of the pretest. Exception included MacWrite, MacDraw, the Lingo Fun foreign language materials, and the Tom Snyder Decisions Decisions materials, all of which were acquired at the request of the workshop participants. In addition, the school system purchased a site license for Logowriter, which replaced Apple Logo for teachers who wanted to use it throughout the system. Furthermore, over 90% of the workshop participants (compared to 50% of the non-participants) reported using MECC software; but this doesn't appear in the table, which is based on a checklist of commercial programs available in the school system.

Insert Table 3 about here.

Finally, it is useful to point out that many of the non-participants were indirectly influenced by the training sessions. For example, English teachers who did not formally participate in the training sessions were impressed with MacWrite and received help from one of the directors and from the other participants to enable their students to use the Macintosh computers as a primary means for teaching English composition. In addition, although only two foreign language teachers participated in the training sessions, the entire foreign language department (five teachers) began to show considerable enthusiasm for using computers. Similar indirect impacts were observed elsewhere throughout the system.

Summary

The training project was enthusiastically received by the participants. In addition, the project clearly seems to have met its objectives.

The unique features which seem to have made this project successful were these: (1) The pre-workshop conferences made it possible to tailor the workshops to the individual needs of the 40 participants. (2) The stipends offered to the participants supplied a gentle incentive to make them want to participate. (3) The availability of time to review software without pressure to complete competing tasks enabled the participants to review a large amount of software in a relaxed and friendly atmosphere that made them want to use the courseware in their classes. (4) The set of 50 disks offered to each participant enabled them to make copies of MECC and public domain software for use in their own classrooms. (5) The combination of hands-on practice and lectures by the instructors provided a stimulating atmosphere in which to review programs and make connections with appropriate curriculum goals. (6) The content modules enabled participants to develop greater expertise in areas in which they wished to specialize. (7) The post-workshop conferences helped the participants to make plans to generalize and integrate what they had learned during the workshops.

This report offers suggestions to enable Crown Point or other school systems to offer similar training programs in the future. In addition, this report suggests activities that can be offered as a follow-up to this project. Finally, this report describes specific types of programs teachers are likely to use both with and without a training program to introduce them to instructional applications of the microcomputer.

Workshop Evaluation Form

Use the red computerized answer sheet to respond to these questions. Use this scale:

a = Strongly agree
b = Agree
c = (In Between)
d = Disagree
e = Strongly disagree

Do not sign the questionnaire. Return it as soon as possible to Helen Jancich.

1. There were sufficient, interesting, relevant programs for me to run and examine. (100% agree)
2. The computer facilities were good. (100% agree)
3. The payment of $35 per session was a satisfactory compensation for my participation. (97% agree; 3% disagree)
4. Four hours was an appropriate amount of time for each workshop session. (94% agree)
5. The pre-planning conference with Dr. Vockell was useful. (94% agree)
6. The follow-up workshop with Dr. Vockell was useful. (94% agree)
7. It is likely that I will be able to apply what I learned during this series of workshops. (100% agree)
8. The one graduate credit hour from Purdue was a strong incentive to encourage me to participate in the workshops. (41% agree; 19% disagree)
9. I would not have participated without the $35 stipend for attending each session. (9% agree; 50% disagree)
10. Dr. Vockell supplied useful guidance and instruction. (100% agree)
11. Helen Jancich supplied useful guidance and instruction. (100% agree)
12. My immediate superiors and supervisors have been supportive of me in taking these workshops. (82% agree; 6% disagree)
13. I know other colleagues who would take this series of workshops if it were offered again. (66% agree; 6% disagree)
14. I would like to take another series of workshops which would build upon skills developed in this series. (91% agree)
15. The opportunity to gain three graduate credit hours by doing additional work was a good idea. (94% agree)
16. The workshop sessions included a satisfactory mixture of theoretical information and practical applications. (94% agree)

Note: Thirty-two persons (out of 40) returned the questionnaires. Because they were anonymous, it was not possible to follow up and obtain 100% responses.

The first number in parentheses after each statement indicates the percentage of respondents either "agreeing" or "strongly agreeing" with each statement. The second indicates the percentage "disagreeing" or "strongly disagreeing" with each statement. When totals do not add up to 100%, the remaining respondents answered "in between."

Table reads: On question 9, 9% of the respondents indicated that they would not have attended without the $35 stipend, whereas 50% disagreed with this statement. (The other 41% were "in between.")

Table 1. Summary of responses to the anonymous evaluation form that participants completed after their post-conference.

Pretest	Non-Participants	Participants
	Elementary Teachers	
(N=92)	(N=51)	(N=15)
(90% Used Computer)	(55% Used Computer)	(100% Used Computer)
Drill & Practice (89)	Drill & Practice (84)	Games (93)
Games (89)	Games (80)	Drill & Practice (80)
Tutorials (75)	Tutorials (47)	Tutorials (67)
Keyboarding (39)	Keyboarding (39)	Simulations (53)
Simulations (37)	Word Processing (37)	Word Processing (47)
Programming (29)	Simulations (29)	Management (47)
Word Processing (23)	Management (18)	Programming (47)
Utilities (12)	Utilities (16)	Utilities (40)
Management (11)	Programming (16)	Keyboarding (33)
Database (8)	Database (8)	Database (20)
Spreadsheet (2)	Spreadsheet (4)	Spreadsheet (0)
	Junior High Teachers	
(N=21)	(N=22)	(N=5)
(48% Used Computer)	(55% Used Computer)	(100% Used Computer)
Drill & Practice (38)	Word Processing (36)	Word Processing (60)
Tutorials (29)	Games (32)	Games (60)
Games (24)	Tutorials (32)	Drill & Practice (60)
Programing (19)	Simulations (32)	Tutorials (60)
Database (15)	Drill & Practice (23)	Simulations (60)
Word Processing (14)	Utilities (23)	Keyboarding (40)
Simulations (14)	Database (18)	Utilities (40)
Utilities (14)	Programming (18)	Programming (40)
Keyboarding (14)	Keyboarding (18)	Database (20)
Spreadsheet (5)	Spreadsheet (14)	Management (20)
Management (5)	Management (5)	Spreadsheet (0)
	High School Teachers	
(N=46)	(N=35)	(N=13)
(35% Used Computer)	(37% Used Computer)	(100% Used Computer)
Tutorials (22)	Word Processing (29)	Games (77)
Word Processing (20)	Tutorials (20)	Word Processing (46)
Games (20)	Simulations (17)	Tutorials (46)
Drill & Practice (20)	Drill & Practice (14)	Simulations (38)
Utilities (13)	Games (11)	Drill & Practice (38)
Management (11)	Keyboarding (11)	Management (38)
Simulations (9)	Utilities (6)	Keyboarding (31)
Keyboarding (7)	Database (6)	Database (31)
Programming (4)	Management (3)	Utilities (23)
Database (2)	Spreadsheet (3)	Spreadsheet (15)
Spreadsheet (0)	Programming (0)	Programming (8)

Table 2. Types of programs run by teachers in 1986 (pretest) and in 1987 by teachers who participated and did not participate in the training sessions. Numbers in parentheses indicate the percentage of respondents who indicated they used each category of programs.

Pretest	Non-Participants	Participants
	Elementary Teachers	
Milliken Math (63)	Print Shop (61)	Print Shop (87)
Print Shop (60)	Certificate Maker (33)	Milliken Math (80)
Apple Logo (45)	Bank Street Writer (33)	Certificate Maker (47)
Bank Street Writer (41)	AppleWorks (31)	Logowriter (40)
EZ Logo (23)	Logowriter (24)	AppleWorks (40)
Crossword Magic (20)	Milliken Math (21)	Bank Street Writer (33)
Newsroom (20)	Stickybear Series (18)	Crossword Magic (27)
Gertrude's Puzzles (19)	Crossword Magic (12)	Microzine (27)
Gertrude's Secrets (19)	Microtype Paws (10)	Stickybear Series (20)
AppleWorks (16)	Gertrude's Puzzles (10)	Microtype Paws (20)
	Gertrude's Secrets (10)	
	Junior High School Teachers	
Print Shop (19)	Print Shop (45)	AppleWorks (80)
Milliken Math (15)	Certificate Maker (41)	Crossword Magic (80)
Bank Street Writer (14)	AppleWorks (27)	Certificate Maker (60)
Crossword Magic (14)	Bank Street Writer (23)	Print Shop (60)
AppleWorks (14)	Crossword Magic (23)	Bank Street Writer (40)
Apple.Logo (10)	MacDraw (14)	Master Type (40)
Master Type (10)	Milliken Math (14)	Milliken Math (40)
Type Attack (10)	PFS: File (9)	Tom Snyder Materials (20)
PFS: File (5)	PFS: Report (9)	PFS: File (20)
PFS: Report (5)	PFS: Write (9)	PFS: Write (20)
	MacWrite (9)	
	High School Teachers	
Print Shop (15)	AppleWorks (17)	Print Shop (85)
Crossword Magic (15)	MacWrite (14)	AppleWorks (69)
AppleWorks (8)	Print Shop (14)	Crossword Magic (62)
Newsroom (6)	Certificate Maker (9)	Certificate Maker (54)
Word Invasion (4)	Logowriter (6)	MacWrite (23)
Master Type (4)	Crossword Magic (6)	MacDraw (23)
Apple Logo (2)	Microtype Paws (6)	Master Type (23)
EZ Logo (2)	Apple Pilot (3)	Lingo Foreign Lang (15)
Gertrude's Puzzles (2)	Bank Street Writer (3)	Tom Snyder Materials (15)
Microtype Paws (2)	MacDraw (3)	Microtype Paws (8)
PFS: File (2)	Juggles Rainbow (3)	Newsroom (8)
PFS: Report (2)	Master Type (3)	PFS: File (8)
PFS: Write (2)	Newsroom (3)	Rocky's Boots (8)
	Lingo Foreign Lang (3)	

Table 3. Top ten programs run by teachers in 1986 (pretest) and in 1987 by teachers who participated and did not participate in the training sessions. Numbers in parentheses indicate the percentage of respondents who indicated they used each software package.

LEARNING MATHEMATICAL CONCEPTS THROUGH LOGO

Sandra V. Turner
Michael L. Land
Department of Computer Education
National College of Education
2840 Sheridan Road
Evanston, IL 60201
(312) 475-1100 x2256

Abstract

The purpose of this study was to investigate the effect of learning Logo on middle-school students' learning of selected mathematics concepts: properties of polygons, estimation of angles and distance, variables, negative numbers, perimeter, symmetry, and rectangular coordinates. Students (n = 91) in the Logo group learned Logo for one hour a week for 16 weeks during the time usually allotted to mathematics, while students (n = 90) in the control group pursued their regular mathematics curriculum without Logo. Students in both groups were pre- and posttested to assess their understanding of the selected mathematics concepts. The Logo group also took a posttest to measure their achievement in Logo. There were no significant differences in mathematics learning between the Logo group and the control group. However, when those who learned the most Logo were compared with those who learned only a minimal amount of Logo, there was a significant difference, in favor of the High Logo group, in their gain in mathematics learning. These findings suggest that students may need to reach a threshold in their understanding of Logo in order for it to be a useful tool for conceptualizing mathematics.

Learning Mathematical Concepts Through Logo

The purpose of this study was to investigate the effect of learning Logo on middle-school students' mathematics learning. Papert (1980) envisioned Logo as a natural and humanistic environment for learning mathematics. He pointed out that too often the mathematics that students learn is dissociated from meaning, and as a result, students learn mathematics by rote. The turtle geometry microworld, however, provides students with an opportunity to make sense of the mathematics that they are learning. It also maximizes students' opportunities for successful learning experiences, helping them to overcome whatever negative self-images they may have about their ability to learn mathematics.

A few researchers (Noss, 1987; Kelly, Kelly, & Miller, 1986/87; Rieber, 1983) have offered evidence to support Papert's thesis that students learn mathematics concepts as they learn Logo, while others (Howe, 1985; Olson, 1985) have found no significant differences in students' mathematics learning as a result of Logo experiences. The first three studies all focused on concepts involving angles and length, which are most closely related to the turtle graphics microworld. This paper reports the results of a study involving a broader range of mathematical concepts: properties of polygons, estimation of angles and distance, variables, negative numbers, perimeter, symmetry, and rectangular coordinates.

Sample

Students in grades 5-8 in four inner-city public schools participated in the study. In two of the schools, students (n = 91) learned Logo for one hour a week for 16 weeks during time normally devoted to mathematics. Students (n = 90) in the other two schools did not learn Logo but received the full allotted time for their regular mathematics curriculum. All four schools were elementary schools in the same urban district with a common mathematics curriculum. In each school ethnic minorities (primarily black and Hispanic) comprised a majority of the student population. The socioeconomic status of the majority of the students was below average.

One of the two Logo schools was a magnet school in which all students scored in the top three stanines on standardized reading and mathematics achievement tests. In this school the study involved one fifth-grade class and one sixth-grade class. Students in the other Logo school consistently scored in the lower six stanines on standardized tests. In the nonmagnet school, sixth through eighth graders participated in the study. In the two control schools the students were sixth, seventh, and eighth graders representing all ability levels.

Prior to the study the Social Sciences Piagetian Inventory (SSPI) (Carter & Ormrod, 1982) was administered to all students. The SSPI is a thirty-item paper-and-pencil test that assesses a student's level of cognitive development from a Piagetian perspective. Scores on the SSPI classify individuals into three broad Piagetian categories: concrete operations, formal operations, or in a transitional stage between these two. The percentages of students classified at the concrete level prior to the study were 48% for Grade 5, 63% for Grade 6, 94% for Grade 7, and 63% for Grade 8. Although the students in the study range in age from 10 to 14 years, there are more seventh and eighth grade students at the level of concrete operations than would be expected on the basis of Piagetian theory.

Procedures

Students in both the Logo and control groups were given pre- and posttests to assess their understanding of the selected mathematics concepts. In addition, the Logo students took a posttest to measure their achievement in Logo.

The goal of the Logo project was to use Logo as an environment for learning mathematics. Logo activities were developed to facilitate the learning of these specific concepts: estimation of angles and distance, properties of polygons, variables, rectangular coordinates, negative numbers, perimeter, and symmetry. These topics were selected because they were part of the middle school mathematics curriculum in the district and because they could be illustrated through Logo. The Logo concepts students learned included iteration, proceduralization, and variables. They planned and debugged their own procedures and modified tool procedures given to them.

Classes were held in a computer lab with two students sharing a computer and working jointly on most activities. The two researchers shared the planning and teaching responsibilities. Each one-hour session included time for both goal-directed activities and exploratory, open-ended activities. In addition to turtle graphics programming, students programmed a robotic turtle for one session. They also experimented with microworlds involving estimation of angles and distance, plotting negative numbers on a number line, and plotting points in the coordinate plane.

Since the sixth-grade students (n = 30) in the magnet school were studying pre-algebra and had participated in a similar Logo mathematics program the previous year (one hour a week for ten weeks), more challenging activities were planned for them. They used Logo to learn about order of operations and evaluating algebraic expressions as well as to do more advanced work with variables.

Students in the control group followed the district's regular mathematics curriculum, which primarily emphasized computation involving whole numbers, fractions, and decimals, but also included a short unit on geometry and measurement. None of the control students had ever been exposed to Logo. About half of them used the computer approximately one hour every three weeks for computer-assisted instruction in mathematics.

Instruments

To assess students' understanding of the mathematics concepts assumed to be related to Logo, the researchers developed a 22-item multiple choice test covering: estimation of angles, estimation of distance and area, properties of polygons, rectangular coordinate system, variables, negative numbers, symmetry, perimeter, and number patterns. The test was piloted the previous year with 110 students in grades 5-8, and several items were revised on the basis of the pilot data. The test-retest reliability coefficient was .76 for a sample of 129 students in grades 5-8 (six-month delay).

To assess students' achievement in Logo, the researchers developed a 20-item multiple choice test. The items on the Logo test asked the students to predict the output of a given Logo instruction, to determine which instruction has the given output, or to fill in a missing instruction in order to produce a given output. The test, which is limited to turtlegraphics, covers iteration, the rule of 360, procedures, and variables, but it does not measure students' knowledge of Logo syntax or the operating system. The Spearman-Brown split-half reliability coefficient for a sample of 51 middle school students was .70.

Results

The following hypotheses were tested:

1. Students in the Logo group will achieve significantly higher on mathematics concepts than students in the control group.
2. Among students in the Logo group, those who learn the most Logo will achieve significantly higher on mathematics concepts than those who learn a minimal amount of Logo.

To investigate the first hypothesis, a one-way analysis of covariance on the mathematics posttest scores by treatment group, using the mathematics pretest scores as the covariate, was performed. The hypothesis was not supported [$F(1,126) = .06$]. Although both the Logo group and the control group improved from pretest to posttest, there was not a significant difference between the two groups as a result of the Logo treatment.

A criterion group design was applied to investigate the second hypothesis. High Logo and Low Logo groups were defined by selecting those who scored more than one-third standard deviation above and below the mean, respectively, on the Logo test. The High Logo group consisted of 11 students who were studying Logo for the first time and 11 who had studied Logo the previous year. The Low Logo group included four students who were in their second year of Logo. Thus, the criterion groups were not related to the amount of time spent learning Logo. A one-way analysis of covariance of the mathematics posttest scores by Logo level, using the mathematics pretest scores as the covariate, was performed. A significant difference, in favor of the High Logo group, was found [$F(1,49) = 5.51$, $p < .02$]. Thus, when we compared those who learned the most Logo with those who learned only a minimal amount of Logo, there was a significant difference, in favor of the High Logo group, in their gain in understanding of the mathematics concepts.

To investigate whether there were any effects due to sex for the two hypotheses, a two-way analysis of covariance was performed for the mathematics posttest scores by treatment group and sex, and by Logo level and sex, using the mathematics pretest scores as the covariate. No significant effects due to sex were found. (All calculated F-values were less than 1.00.)

Discussion

The purpose of this study was to investigate the effect of learning Logo on middle-school students' knowledge of selected mathematics concepts. There was no significant difference in the mathematics learning of those students who studied Logo and those who studied the regular mathematics curriculum. However, when we compared those who learned the most Logo with those who learned only a minimal amount of Logo, there were significant differences, in favor of the High Logo group, in their gain in mathematics learning.

These findings suggest that students may need to reach a threshold in their understanding of Logo in order for Logo to be a useful tool for conceptualizing mathematics. A similar observation has been reported by Noss (1984). Noss, whose work was based on an unstructured, exploratory approach that encouraged children to pose their own programming projects, observed that many children reached plateaus in their programming, both in their unwillingness to pose problems which demanded new ideas and in their reluctance to explore unfamiliar territory. Our study suggests that overcoming such plateaus may be an important prerequisite to using Logo as a tool for learning mathematics.

References

Carter, K. R., & Ormrod, J. E. (1982). Acquisition of formal operations by intellectually gifted children. *Gifted Child Quarterly*, 26, 110-115.

Howe, J. (1985, March). Logo programming and intermediate mathematics. *Computers in Education*, pp. 7-10.

Kelly, G. N., Kelly, J. T., & Miller, R. B. (1986/87). Working with Logo: Do 5th and 6th graders develop a basic understanding of angles and distances? *Journal of Computers in Mathematics and Science Teaching*, 6(2), 23-27.

Noss, R. (1984). Explorations in mathematical thinking: Some implications from Logo classrooms. *Pre-Proceedings of the 1984 National Logo Conference*. Cambridge, MA: Massachusetts Institute of Technology, pp. 85-92.

Noss, R. (1987). Children's learning of geometrical concepts through Logo. *Journal of Research in Mathematics Education*, 18(5), 343-362.

Olson, J. K. (1985). Using Logo to supplement the teaching of geometric concepts in the elementary school classroom. Unpublished doctoral dissertation. Oklahoma State University.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.

Rieber, L. R. (1983). The effect of Logo on increasing systematic and procedural thinking according to Piaget's theory of intellectual development and on its ability to teach geometric concepts to young children. Unpublished Master's thesis. University of New Mexico.

GRAPHICAL TECHNIQUES AS REVISION AIDS: AN EXPLORATION WITH LOGO

Pamela Freyd
Kathe Morris-Basik
Graduate School of Education
University of Pennsylvania
3700 Walnut Street
Philadelphia, PA 19104
(215) 898-5690

ABSTRACT

Graphical Techniques as Revision Aids: An Exploration with Logo.

The future of writing is bound to computers. From talking textwriters to integration with graphics and spreadsheets, from revision aids to desk top publishing, technology is giving us new tools with which to write. These new tools present new challenges to writing teachers. They raise questions about how people learn to write and think - questions which must be addressed before writing instruction can enter the computer age: how do computers affect the writing processes of student writers? What sort of writing instruction does new technology require? Can we integrate computers into existing writing programs, or do new, high technology programs need to be developed? What are the types of electronic tools that are the most helpful to developing writers?

Early visionaries predicted that students would write better if given access to computers and word processing; research, however, has shown that the situation is not so simple. Students may write more and the work may be better edited, but computers alone do not lead to qualitative improvement in student writing. Asking students to revise written work has generally resulted in changes in the surface-level features of spelling, punctuation or word (Daiute, 1985) rather than in changes of global features. The focus of this research is exploration of electronic tools that might be helpful to young writers to step back and see their writing at a global rather than word level.

This paper reports on a pilot project designed to explore the possibility that graphical representations might facilitate revision in 6th grade students. We used the Logo programming language because it was a computer use with which our subjects were already familiar.

When children work to debug a Logo program, they do so because the picture that they see does not match the picture that they intended to make or because they note some way to improve it. It is easy to think of a picture in a global fashion; it is not perceived linearly as is writing. When we try to help children learn to revise compositions, we try to get them to see their writing at a global level, to step back and see the "picture" of what they have written. It seems reasonable, then, that finding graphic analogies to writing might help students see writing in a more global fashion.

Two teachers alternated in the presentation of lessons over a period of three months to a Logo Connection Group (7 children) and a Logo Only group (6 children). Both groups spent the same amount of time working at the computer with Logo. but the activities of the Logo Connection group were specifically aimed at relating Logo activities to writing tasks involving revision.

Quantitative analysis showed that students in the Logo Connection group made more additions as they revised the prepared exercises. A qualitative analysis by three researchers showed that students in the Logo Connection group made more revisions at the global level than the Logo Only group.

Ongoing study is extending this research to graphical abstraction of text analysis to help students at 4th and 6th grade levels whom teachers identify as having problems with written sentence structure (e.g. run-on sentences).

In normal conversation, a speaker who perceives that an audience is not reacting as expected might respond by repeating what has been said, by rephrasing, by adding more information or even by yelling. This is not the case with written language. The job of predicting audience reaction to written text rests with the writer who does not have benefit of feedback. *Revision*, the process of reading text, comparing it with some internalized idea of audience reaction and then changing the text in an attempt to improve it, is clearly a complex task. Revision requires thinking about one's own thinking and monitoring one's own progress, skills that seem to develop relatively late in children (Brown, 1975). It is not surprising, therefore, that children do not have skilled revision strategies.

Complicating the issue of revision for children has been the sheer physical labor involved in rewriting text. Because children are also busy mastering physical writing skills, spelling skills and editing skills, it has been difficult for teachers to focus on serious revision instruction with elementary age children. For children, improving written work has generally meant attending to surface level features of spelling, punctuation or word changes (Daiute, 1985). The most successful instructional strategies in the area of revision have been those that involve conferencing with peers and teachers (Graves,1983; Freedman, 1981; Calkins,1980). Conversations with others about their work seem to help children become aware of their own writing processes and help them to see and evaluate what they have written in terms of their own goals.

Technological developments have the potential to change the ways that teachers help children learn to write and learn to "shift between reader and writer, between critic and creator" (Calkins, 1980 p 334). The emerging integration of word processing, spelling checkers and text editing programs in writing curricula gives us the opportunity to rethink teaching strategies. Word processing facilitates the manipulation of text without the tedium of recopying. It makes changes and corrections easy and allows alternative arrangements of text. Spelling checkers have the potential to refocus traditional concern over correct spelling and allow children to put more thought into the content of the writing. Text editors may free children from anxiety at the surface editing level and allow for greater concentration in matters of organization and style. Certainly these new tools appear to support writing processes as we have come to understand them (Emig, 1971; Perl, 1979; Flower & Hayes, 1981; Graves, 1983; Bereiter & Scardamalia, 1985; Daiute, 1985). It is important to remember that the needs of developing writers are very different from those of skilled writers. Writing tools must be carefully designed for use by children (Pea & Kurland, 1986).

Albert Molner, in a talk at the NEEC'87 Convention in Philadelphia noted that "doing things better is not the same as doing better things." The history of technological development is full of examples of new inventions that are designed to do an existing task in a better way but then the invention is used to do better things, tasks that could not be done before. Computers themselves, although designed to manipulate numbers, have found far-reaching use in word manipulation. Word processing and the related writing aids that are rapidly being developed will let many people do a better job of writing following the traditional strategies. Computer programs have been designed that simulate the dialog of a writing conference. On-line planning and revision programs are available. Perhaps computer technology will let us develop new and possibly better strategies for helping writers' revise. The computer's ability to represent various aspects of writing graphically may be such a technique (Perlman & Erickson, 1983).

Word processing has replaced the typewriter in the business world. Most professional writers rave about its benefits. With the appearance of low cost microcomputers in the early 1980's, many people began to see the educational potential of word processing. Early visionaries predicted immediate improvement in children's writing and revision skills. It hasn't been so simple. Research has shown that young people don't just begin to revise when given access to computers and word processing technology (Daiute, 1985). They need help and guidance, but (at least junior high students) can be taught to revise using open-ended on-line questions that help them to reflect on whether the actual text matches what they intended to convey (Daiute, 1985). Even high school and college students need some stimulus to begin to take advantage of word processing power and to revise the meaning or form of their texts rather than change only punctuation or words (Harris, 1985; Kurth, 1987). A great deal of computer software is currently being developed to help writers both to plan and then to reflect on their completed texts (e.g. *Writer's Helper* by William Wresch).

Research studying children's writing processes using computers must necessarily proceed cautiously and slowly. One must consider whether there are enough computers available to have a significant impact on writing instruction. Students must have sufficient access to these computers to make their use meaningful. Both teachers and students need time to become comfortable with the equipment and to let the novelty wear off. Using computers for writing requires not only keyboard familiarity but also familiarity with the hardware and software. These requirements must be met before we can be sure that we are looking at writing processes and not other factors. At the same time, there is need to experiment with and explore techniques as computers are entering schools and as programs are being developed. This paper reports on one pilot project in which we set out to explore the possibility that graphical representations might facilitate revision skills in children.

We used the Logo programming language because it was a computer use with which our subjects were already familiar and comfortable since they had been using it for three years. Logo is a computer language designed for use by children. It is very easy to begin to work in Logo while at the same time it extends to highly advanced computer programming. Most people are acquainted with Logo through its graphics capabilities. If our instructional techniques seemed successful, they might be appropriate candidates for inclusion in computerized developmental writing programs after refinement and large scale testing.

There are many similarities between computer program writing processes and composition writing processes: prewriting, composing and revising (English & Edwards,1984; Newkirk, 1985; Freyd, 1987). Because early work in Logo is highly graphical, children get feedfack quickly when they try to run their programs. Graphic goals may have a particular salience for children because they get a "picture" which either matches their goal or not. Papert has made a strong claim that children will learn to think about their thinking (metacognition) by programming in Logo (Papert, 1980). Short term studies looking at the transfer effects of Logo on problem solving skills have been inconclusive (Pea & Kurland, 1984). There is currently debate in the literature over just what components one ought to be looking for and how one should proceed in the area of educational computer research (Becker, 1987; Papert, 1987; Pea, 1987; Walker, 1987). There is some hint that metacognitive skills are improved through using Logo (Clements, 1986). It is not known what the long term low level effects on childrens attitudes, strategies and cognitive skills work with computers or Logo may eventually have (Perkins, 1985).

It is clear, however, that in order to get a Logo program to run, debugging (or revision) is generally necessary. As in composition work, it seems that children do not bring good strategies to the task but rather need guidance in developing them (Carver & Lahr, 1986). Because the students with whom we were working had been using Logo for over three years, we operated on the assumption that if there were any long term low level effects of Logo on revision processes in general we might expect to see them. We made this assumption because the students were able to write both longer and more complicated programs at the end of three years. In order to do this we assumed that they had been developing more sophisticated or successful debugging or revision skills. We hypothesized that direct instruction might result in some transfer of these skills to revision work in composition.

When children work to debug a Logo program they do so because the picture that they see does not match the picture that they intended to make or because they note some way to improve it. It is easy to think of a picture in a global fashion; it is not perceived linearly as is writing. When we try to help children learn to revise compositions, we try to get them to see their writing at a global level, to step back and see the "picture" of what they have written. Beach (1976) in a study of college junior and seniors noted that students who revised significantly seemed to "conceive of the paper in holistic terms." (p 162). Inexperienced revisers, on the other hand, focus on changes at the word or sentence level. Hayes et al. (1985) determined that expert revisers reviewed their work as a whole-text task. They created a global goal to guide their processes.

It seemed reasonable to us, then, that finding graphic analagies to writing might help students see writing in a more global fashion. We know that graphic techniques of charts and graphs help people better interpret numerical information. Bernhardt (1986) has argued that encouraging students to experiment with visible features of text can change the ways people conceive classificatory relations. Dixon (1985) has suggested that juxtaposing symbol systems can improve thinking skills. Our purpose is to explore graphic representations and abstractions as potential facilitators to helping writers step back to view their writing. If our techniques seemed successful, they might be appropriate candidates for inclusion in computerized developmental writing programs.

Students:

This pilot project, one of several involving graphical representations and abstractions related to writing, explores some techniques using Logo that might be conducive to helping children become more aware of and gain greater control of revision strategies in composition writing. All but one of the thirteen sixth grade students had been exposed to Logo for three years in weekly computer lab classes. The students had also been introduced to word processing through Bank Street Writer, but because the student-computer' ratio was 26:1 and because computers were not yet included in the language arts curriculum, they were not used regularly for word processing in school work. Five students had computers at home and of these, three used them for word processing. Although a limited resource, computers were not novel to these children.

The students attended a racially and economically heterogeneous K-8 public school in downtown Philadelphia. They were all performing well at their grade level and were considered to be in the top third of the 95 students in the grade as rated by a combination of standardized test scores and teacher judgment. These children were selected for this project by their language arts teacher because she felt that they would benefit from extra help with their writing while at the same time be able to cope with any material missed when they were out of the regular classroom. Two students from the racially mixed group spoke English as a second language but were now considered by the school to be proficient in English. Our small group had a balanced racial composition, but there were more boys (8) than girls (5). We met with the children for 14 half hour sessions during the spring of 1987. We worked in a computer lab which had a combination of Apple 2e and Commodore 64 computers.

There is a strong focus on process writing within these student's classroom and within the school as a whole. The bulletin boards in the hallways are filled with samples of children's work. Students in the school are frequest winners of writing contests. The children's experienced and creative teacher regularly attended seminars and workshops to improve her expertise in teaching writing. There was support and interest for our work with the students.

We surveyed the students to learn how they felt about writing and computers. Even after three and a half years of computer instruction in Logo, most students associated computers with playing games. They generally expressed positive attitudes towards writing, but noted that it was extremely difficult to begin to write, to find something to say (Bereiter & Scardamalia, 1987). They very much disliked having teachers tell them that something that they had written needed to be redone. These students seemed typical of students described in the literature.

Method:

We let the students divide themselves into two groups because we wanted them to feel as comfortable as possible in the condition of a pull-out program with new teachers. Our only restriction to the students was that the groups couldn't be either all girls or all boys. We hoped that working with friends would encourage confidence and willingness to express themselves. Each group had a work location in the lab and the two teachers alternated weekly between the groups. With the flip of a coin on the first meeting, one group became the Logo

Connection group (7 children) and the other the Logo Only group (6 children). All the children spent the same amount of time working at the computer with Logo. The activities of the Logo Connection group, however, were specifically aimed at relating Logo activities to the writing task while no connection was made for the Logo Only group.

Lessons followed a set format: first we introduced a Logo activity either as a demonstration or an exercise on which the children worked at the computer with a partner for about ten minutes; then there was a guided discussion before the children worked on revising a paragraph or story that we provided. Revision assignments were given on triple space computer printouts and students were directed to improve the writing using pencil or pen on the paper printout. Students had about twenty minutes to work on the revisions although they usually finished in twelve to fifteen minutes. One subject who seemed to work more slowly than the others often stayed an extra five or ten minutes to complete his work. Most sessions were taped so that we could record children's comments both during discussion and while revising.

During the four months of our visits, the students worked on eight assignments that we asked them to improve. We examined the writing curriculum used in instruction of the students and picked out areas in which they had worked such as sequence, character, setting, theme, etc. Since we wanted students to focus on meaning and style, activities did not contain mistakes in spelling and punctuation. Exercises could be revised in many ways: combininbing short choppy sentences, restructuring chronology, changing the plot or character or setting, strenghtening arguments, changing for a different audience, or changing the type of writing. For example, in one session we presented the children with a short news story of a sports event. Our instructions were to change the focus of the story so that it might appear in the feature section of a paper. The preliminary computer work for the Logo Connection group involved exploring a simple program that created a design. The exploration focused on changing the angle of a turn in the program and noting its effect on the nature of the design that resulted. Children observed that a single change in the angle could result in a very different picture. The guided discussion then led to a comparison between changing the angle in a Logo program and changing the angle of a story. The Logo Only group also explored designs, but discussion was based on the questions that the students raised and no parallel to writing was made. The difference between the groups involved the direct reference to the Logo work. Instructions for revision were identical.

(Computer demonstration - angle lesson.)

Although we did not set out to look at Logo teaching techniques, both researchers made the same observation in their notes: the Logo Connection group became far more involved in computer use. When students had completed their revisions, they could go back to the computers if they wished and work during some of their recess time. While there are undoubtedly many possible reasons for this involvement, it was indeed the case that Logo Connection students had to be pulled away from the computers so that they could get at least some of their recess time. The Logo Only group was generally ready to leave with the bell. At this time, we have no reason to think that there was an intrinsic difference difference between the children in these groups but rather that the difference was in the workshop atmosphere created by the directed analysis of the Logo programs for the Logo Connection group as opposed to the less directed self-discovery of the Logo Only group approach.

All the children seemed to take a serious and involved interest in the written work. They expressed no problems working with a text they didn't write, although they did express a liking for some of the topics more than others. They seemed to get in the spirit of the activity and find some satisfaction in their editorial roles. Some of their interest appeared to grow from their sense of ownership of the writing as indicated by their opinions of the quality of the changes. Our evidence in this respect was their eagerness to read the pieces that they had revised aloud to their friends and to us.

For several reasons we chose to work with teacher prepared passages rather than with the students' own work: the passages gave us common ground for analysis; out limited time precluded what we felt were appropriate prewriting and writing sessions; and the experimental situation did not seem conducive to rapport required for critical self-examination of work. We decided that revisions made on printouts rather than computer were preferable at this time because there was too much variability in the students familiarity with the keyboard and with word processing commands. For most exercises, we typed in the changes that the children made and provided copies of all revised work for all children.

Evaluation:

We chose a simple method of analysis that seemed best suited for the task that we had designed. We looked first at the quantity of the revisions and then at the quality of the revisions. We hypothesized that students who were actively considering alternatives and experimenting with text might make more revisions in absolute terms. We also hypothesized that the Logo Connection group would make more revisions affecting the quality of the exercises, revisions that could be categorized at a global level.

Analyzing writing and categorizing changes is challenging. We based our analysis on the Faigley and Witte (1981) model that differentiates *surface* *changes*, including formal changes (spelling, tense, etc.) and meaning-preserving changes in (additions, deletions, substitutions, etc) from *meaning changes* at either the micro or macrostructure level (additions, deletions, substitutions, etc.). See Table 1. Global changes are those changes that would result in a change in the summary of a text. An analysis of of mature and immature characteristics such as the presence of complex vs simple sentences (Sharples (1985); Wilkinson et. al. 1980) did not seem appropriate since we supplied the original text.

(Insert Table 1 - overhead)

Papers were quantitatively coded by the two researchers with a 94% consistency level. Results are presented in Table 2.

(Insert Table 2 - overhead)

In the category 'additions' the Logo Connection group showed a trend of greater revision activity than the Logo Only group . In other categories both groups performed essentially the same. We had expected that the 'addition' category might contain the greatest number of changes for all students since a typical way for children to make changes is to add new information (Bereiter & Scardamalia, 1987). That this is not the case is possibly the result of the fact that children were not working with their own texts and that their instructions for revision had a focus other than adding new information. We did not find examples of the categories of 'distributions' and 'consolidations' that were included in the Failgey & Witt model. We tentatively assume that these are, perhaps, examples of more mature manipulations of written language than our subjects used at this time.

The quantitative analysis indicated only that the Logo Connection group seemed to make more additions than the other group. What is of greater interest to us was the effect of those revisions on the total text. For example, changing the age of a character might result in many revisions that necessarily followed in various places in the story. Many small changes often result in a single global change.

Revision is, afterall, about meaning and rhetorical effect. We wanted to use measures that were appropriate both to the task and to the understanding that the children brought to the task. Thus, we examined the writing curriculum used in the instruction of the children and noted the areas with which we could expect them to be familiar. These categories were: genre, plot, chronology, character, setting, theme, figurative language, irony, and sentence structure, the areas that were in our minds when we prepared the assignments. The 104 exercises were read by three evaluators with instructions to code the pieces for global changes in these categories and to note any categories that they felt should be added. One evaluator had no knowledge of the project. A change was coded if it was felt that it would result in a change in the summary of the piece or if the rhetorical effect was altered. Although the coders varied slightly in the number of changes that they attributed to the pieces, the rank order of the students remained the same. The results are presented in Table 3.

(Insert Table 3 - overhead)

Five of the seven subjects in the Logo Connection group were among the higher scoring subjects.

It appeared that students in the Logo Connection group were making a greater number of global type revisions. We wondered if this was the result of different talents that the students brought with them to the project. Perhaps some students were more prone to revising and the differences we saw had nothing to do with our work. To get a baseline of students' revision styles, we analyzed two examples of their own writing and revisions. One was done at our request before the beginning of our experiment in the students own classroom; the other was done with us in the middle of our project. Directions for both writing exercises were the same: the writing was done in one full class setting and the revisions were made at another session using a different color pen or pencil. The in-class writing was a creative writing assignment using a sentence completion prewriting activity. The assignment for the lab was a persuasive letter designed to convince the principal to increase recess time. Unfortunately, three students were absent on the day that the in-class writing assignment occurred. Results, therefore, are for the ten remaining subjects. See Table 4.

(Insert Table 4 - overhead)

There are no significant differences between the groups. The great number of deletions in the Logo Connection Group were due to one student who completely rewrote his piece. Students showed a far greater variation in the amount of revising their own writing than they did during the exercises. This is probably to be expected given the different nature of the tasks. A qualitative analysis of children's own work showed very few changes. See Table 5. Four of the five students in the Logo Connection Group were among the high ranking revisers.

(Insert Table 5 - overhead)

There were a total of 5 global changes in the Logo Connection group and a total of three changes in the Logo Only group. From these samples of the student's work, it does not appear that the subjects in the groups differed significantly in the skills and approaches that they brought to the revision task.

We recalculated the quantitative analysis using the ten remaining subjects. See results in Table 6. The patterns are the same as for the larger group.

(Insert Table 6 - overhead)

Conclusions

The results of this pilot project seem to indicate that graphical representations of some aspects of writing may have the potential to encourage students to think about and to do more revising. We would be reluctant to make any generalizations, however, given the small number of students involved. Nor do we feel that the technique involving Logo as described in this paper is necessarily the best or only direction to proceed. For us, it was a starting point given the nature of the school situation that we encountered. We are currently embarking upon a year long study examining similar exercises in terms of the effects on the subjects own writing within the context of a curriculum that now includes regular use of word processing. In addition, we are examining different graphic techniques based on computational text analysis (Perlman & Erickson, 1983) to help students see what their teachers feel are problems with their sentence structure (in particular run-on sentences).

We feel that we have embarked on an fruitful research endeavor, one that has the potential for interesting results both at the theoretical and at the practical level. To date, most computerized writing aids such as spelling checkers and style checkers focus only on surface level features of language. Open ended questions for prewriting and revising do offer a more global focus. We believe, however, that it is important to explore new techniques that the computer makes possible for visualizing text. The results of this pilot study have encouraged us to proceed with further exploration in the area of graphical representations and abstractions as they may help students to step back to to view their writing as a whole, to help them gain the perspective needed for self-evaluation.

BIBLIOGRAPHY

Beach, R. (1976). Self-evaluation strategies of extensive revisors and nonrevisors. *College Composition and Communication.27*, 160-164.

Becker, H. J. (1987). The importance of a methodology that maximizes falsifiability: Its applicability to research about Logo. *Educational Researcher, 16(5), 11-16.*

Bereiter,C. & Scardamalia, M. (1987). *The Psychology of Written Composition.* Hillsdale, NJ: Lawrence Erlbaum.

Bernhardt, S. (1986). Seeing the Text. *College Composition and Communication, 37,1,* February, 66-78.

Bridwell, L. & Duin, A. (1985) Looking in depth at writers: computers as writing medium medium and research tool. In J. Collins & E. Sommers (Eds.) *Writing On-Line* Boynton/Cook. 142-150.

Brown, A. L. (1975). The development of memory: Knowing when, where, and how to remember. In R. Glaser (Ed.), *Advances in instructional psychology.* Hillsdale, NJ: Lawrence Erlbaum.

Calkins, L.M. (1980) Children's Rewriting Strategies. *Research in the Teaching of English* pp 331-341.

Calver, S. & Lahr, D. (1986) Assessing Children's Logo Debugging Skills with a formal model. *Journal of Educational Computing Research 2(4).* 487-525.

Clements, D.H. (1986). Logo and cognition: a theoretical foundation. *Computers in Human Behavior,2.* pp 95-110.

Collier, R.M (1983). The word processor and revision strategies. *College Composition and Communication,34*, 149-155.

Daiute, C. (1985). Do writers talk to themselves? In S. Freedman (Ed.), *The Acquisition of Written Language: Response and Revision*, Norwood, NJ: Ablex.

Daiute, C. (1985). A self-question strategy to increase young writers' revision processes. *Applied Psycholinguistics 6*, 307-318.

Dixon, W. P. (1985). Thought-provoking software: Juxtaposing symbol systems. *Educational Researcher*, May, 30-38.

Emig,J. (1971). The composing process of twelfth graders. Urbana,IL:National Council of Teachers of English.

English, R. & Edwards, G. (1985). Programming as a writing activity. *The Computing Teacher*, February, 46-47.

Faigley, L. & Witte, S. (1981) Analyzing revision. *College Composition and Communication,* 32, 400-414.

Flower, L. & Hayes, J.R. (1981). A cognitive process theory of writing. *College Composition and Communication*, 32(4), 365-388.

Freedman, S. (1981). Influences on evaluators of expository essays: Beyond the text. *Research in the Teaching of English, 15*, 245-255.

Freyd, P. (1987). Bugs in the Workshop: Using Logo to Enhance Language Teaching. (Paper presented at the national Educational Computing Convention, Philadelphia, PA, June). Available in *NECC'87 Proceedings.*

Graves, D. H. (1983). *Writing:Teachers and children at work.* Exeter, NH:Heinemann.

Harris, J. (1985). Student writier and word processing: A preliminary evaluation. *College Composition and Communication, 36*, 323-330.

Hayes J.R., Flower,,L.S., Schriver,K., Stratman,J., & Caey,L. (1985). Cognitive processes in revision (Tech.Rep.No.12). Pittsburgh, PA:Carnegie Mellon University, Communications Design Center.

Hertz, R. (1983). Problems of computer-assisted instruction in composition. *The Computing*

Teacher. September, 62-64.

Kiefer,K. & Smith, C. (1984). Improving students' revising and editing: The Writer's Workbench System. In W. Wresch (Ed.) *The Computer in Composition Instruction* pp. 65-82. Urbana, IL:National Council of Teachers of English.

Kline, C. H. (1983). The influence of Logo revision strategies on the written revision practices of young children: A stepping stone. ERIC ED270743.

Kurth, R. (1987). Using word processing to enhance revision strategies during student writing activities. *Educational Technology* Jan. 13-19.

Land, R. E. (1984). Revision strategies of seventh and eleventh graders. ERIC ED261410.

Newkirk, T. (1985). Writing and programming: Two modes of composing. *Computers. Reading and Language Arts*. Winter, pp 40-43.

Papert, S. (1980). *Mindstorms: Children Computers and Powerful Ideas*. New York:Basic Books.

(1987). Computer criticism vs. technocentric thinking. *Educational Researcher*, 16(1), 22-30.

Pea, R. & Kurland, M. (1984). Logo programming and development of planning skills. (Tech. Rep. No. 16). New York: Bank Street College of Education, Center for Children and Technology.

Pea, R. & Kurland, M. (1986) Cognitive technologies for writing. In *Review of Research in Education 13*

Pea, R. The aims of software criticism: Reply to Professor Papert. *Educational Researcher, 16(5)*, 4-8.

Perkins, D.N. (1985) The fingertip effect: How information-processing technology shapes thinking. *Educational Researcher* Aug/Sept. 11-17.

Perl S. (1979) The composing processes of unskilled college writers. *Research in the Teaching of English,13(4)*, 317-336.

Perlman, G. & Erickson, T. (1983) Graphical abstractions of technical documents. *Visible Language*, Autumn, 380-389.

Sharples, M. (1985). *Cognition, Computers and Creative Writing*. New York:Wiley.

Sommers, N. (1980). Revision strategies of student writier and experienced adult writers. *College Composition and Communication, 31*, 378-388.

Walker, D. F. (1987). Logo needs research: a response to Papert's paper. *Educational Researcher, 16(5)*, 9-11.

White, M. A. (1987). Information and imagery education. In M. White (Ed.) *What Curriculum for the Information Age*. New York: Teachers College Press.

Wilkinson, A., Barnsley, G., Hana, P., & Swan, M. (1980). *Assessing Language Development*, Oxford: Oxford University Press.

Table 1. Faigley & Witt (1980) Classification of revision changes.

I.Surface Changes A. Formal Changes
- 1. Spelling
- 2. Tense, number, and modality
- 3. Abbreviations
- 4. Punctuation
- 5. Format
  - a. Paragraph
  - b. Other

B. 8Meaning-Preserving Changes
- 1. Additions
- 2. Deletions
- 3. Substitutions
- 4. Permutations
- 5. Distributions
- 6. Consolidations

II. Text-Based Changes

A. Microstructure Changes
- 1. Additions
- 2. Deletions
- 3. Substitutions
- 4. Permutations
- 5. Distributions
- 6. Consolidations

B. Macrostructure Changes
- 1. Additions
- 2. Deletions
- 3. Substitutions
- 4. Permutations
- 5. Distributions
- 6. Consolidations

Table 2 **Changes at Word Level**

Mean number of revisions at word level for each subject based on 8 samples

	Logo Connection Group n=7	Logo Only Group n=6	
additions	x = 18.07	x = 11.13	t = 1.60
	s = 9.88	s = 5.39	df 11
deletions	x = 15.18	x = 13.22	n.s.
	s = 7.16	s = 9.64	
substitutions	x = 17.53	x = 18.48	n.s.
	s = 5.74	s = 7.59	
permutations	x = 2.43	x = 3.73	n.s.
	s = 1.53	s = 2.58	

Table 3 *Results of qualitative evaluation of 13 subjects each with 8 writing samples*

group **L=Logo Only group** **LC=Logo Connection group**	**# global revisions**	**subject**
L	16	KJ
LC	16	PS
LC	14	AZ
LC	13	GJ
LC	13	DL
LC	12	KC
L	11	LC
L	11	CN
L	11	SS
LC	10	MO
L	9	KM
L	9	NB
LC	8	CD

Table 4

Mean number of word level changes in two samples of students own work

	Logo Connection Group n = 5	**Logo Only Group** n = 5	
additions	x = 8.40	x = 12.80	ns
	s = 10.59	s = 8.06	
deletions	x = 15.30	x = 5.00	ns
	s = 23.53	s = 7.85	
substitutions	x = 4.40	x = 6.40	ns
	s = 4.89	s = 9.21	

Table 5
Results of qualitative evaluation of 10 subjects including own writing samples

group L=Logo Only group LC=Logo Connection group	8 writing samples exercises # global revisions	subject	2 writing samples own work # global revisions
LC	16	PS	1
LC	14	AZ	1
LC	13	GJ	0
LC	12	KC	2
L	11	LC	0
L	11	CN	2
L	11	SS	1
L	9	KM	0
L	9	NB	0
LC	8	CD	0

Table 6

Changes at Word Level
Mean number of revisions at word level for each subject based on 8 samples

	Logo Connection n = 5	Logo Only n = 5	
additions	x = 18.55	x = 9.53	t = 1.88
	s = 9.90	s = 4.14	p > .10 df 4
deletions	x = 14.34	x = 10.80	
	s = 8.54	s = 8.50	ns
substitutions	x = 18.40	x = 17.93	
	s = 3.34	s = 8.35	ns
permutations	x = 2.39	x = 4.08	
	s = 1.70	s = 2.73	ns

LEARNER CHARACTERISTICS AND LOGO PROGRAMMING: MATCHING STUDENTS WITH TEACHING METHOD FOR EFFECTIVE INSTRUCTION

S. Kim MacGregor
Judith L.Repman
111 Peabody Hall
Louisiana State University
Baton Rouge, LA 70803

Abstract

The purpose of this research was to explore the relationship between learner characteristics, instructional methodology, LOGO programming performance, and understanding of the concept of variable. One hundred five students were assigned randomly to one of three methods of programming instruction (teacher structured, teacher guided, and student discovery). Before instruction commenced, students were tested to obtain measures of mathematics achievement, locus of control (internal or external), cognitive style (field independent or dependent), creativity, and self esteem. Students received eight weeks of LOGO programming instruction, and at the end of the eight weeks took tests measuring programming performance and understanding of the concept of variable. The results of the data analyses indicated that gender, cognitive style, and prior mathematics computational ability have a significant effect on programming performance. Three interactions were also found to effect programming performance. The significant interactions included gender by treatment, locus of control by treatment, and self esteem by treatment. In addition, method of instruction was found to have an impact on students' understanding of the concept of variable. In general, the most effective methodology was the teacher guided approach whereas, the least effective was the student discovery approach. The results are discussed with respect to their implications for practitioners in instructional settings.

Introduction

Educators and researchers have investigated, formally and informally, the relationship between certain learner characteristics and academic achievement. It is known that gender, socioeconomic status, age, race, and school characteristics are demographic variables related to student achievement. Increasing awareness of and interest in cognitive psychology has led to systematic research on different aspects of learner characteristics--cognitive and affective variables--and their effects on learner performance [1]. Cognitive and affective variables that can be considered include developmental stage, cognitive style, locus of control, creativity, self-esteem and motivation [2]. One proposed avenue of research recommends analyzing differences between successful and unsuccessful learners to understand which of these differences may significantly affect learning [1]. While educators have investigated the effects of demographic and cognitive factors on student performance in traditional subjects, only in recent years has the content area of programming been considered.

Programming as Instructional Context

An entire computer language has been created to treat programming as a means to get to a creative and precisely defined end. LOGO is an interactive and procedural language specifically designed as a learning environment. Seymour Papert, creator of LOGO, intended the language to be an environment in which the learner could explore, discover, solve problems, and create such that process is considered to be as important as final product [3,4]. It is believed that the process may be enhanced by the learner's cognitive style and creativity. Creativity is defined in terms of personality traits including self-confidence, risk taking, self-sufficiency, introversion and cognitive style [5,6]. A question of importance in this study is whether LOGO instruction is equally effective for all students or whether students with certain characteristics are more likely to benefit from programming experiences. In addition, the issue of whether differences in programming performance exist as a result of the interaction between instructional methodology and certain characteristics is explored.

Importance of Prerequisite Skills

In their review of the literature concerning the demands and requirements of computer programming Dalbey and Linn [7] found few studies on the psychological or educational aspects of computer programming. Since computer programming is a relatively new addition to the pre-college curricula, most of the investigations of the relationship between cognitive style and programming success have been conducted with college students [8]. In a study involving high school students enrolled in a BASIC programming course, Pommersheim and Bell [9] found a significant effect for students who had a Type I cognitive profile. A student with a Type I profile exhibited at least four of the following cognitive styles: field independent, focuser, narrow, complex, reflective, sharpener, or tolerant. While gender effects have also been identified by several studies, the research is inconclusive on the relationship between gender and success in computer programming environments [8,9,10]. In a study that involved LOGO programming and early elementary students, Bradley [11] found that a top-down information processing style correlated significantly with field independence and holistic processing. Successful LOGO programming was positively correlated with top-down processing and academic achievement. It is apparent that there is a relationship between the cognitive characteristics of the learner and programming performance.

Impact on Learner Outcomes

The rationale for teaching computer programming to pre-college students is often based on the assumption that certain programming languages serve as a conceptual framework for the development of cognitive abilities and the understanding of mathematics [3,12]. Research results on the cognitive effects and achievement outcomes of LOGO programming have been mixed. One of the most significant obstacles in the assessment process is the lack of agreement on appropriate measures to assess the outcomes and cognitive consequences of LOGO programming instruction [13,14]. Extensive studies reported by Pea et al. [15] have failed to show that learning LOGO programming has led to skills transfer in the area of planning. At the same time, studies by Clements [16,17] have shown a positive effect for LOGO programming on certain metacognitive and cognitive abilities as well as creativity. The inconsistencies in existing findings may be a result of the attempt to measure very general effects in terms of problem solving and mathematics achievement. A "micro approach" to the investigation of transfer of learning suggests looking at a very specific skill. The current research addresses the question of whether understanding of the specific concept of variable transfers to non-computer tasks. The concept of variable is important in that it is one of the few central themes which serves to organize, unify, and give coherence to mathematics at all levels. It is believed that LOGO may be used to provide a concrete model and direct application of this concept [3].

Importance of Instructional Method

Seymour Papert envisioned the LOGO programming environment as a microworld where students would come to a deeper understanding of mathematical concepts and problem solving strategies through a process of discovery [3]. Research in the area of thinking skills and problem solving has generally concluded that direct instruction is the most effective (and for some skills, the only) way to motivate students to use cognitive and metacognitive skills and strategies in their own thinking and problem solving [18,19]. Utilization of the student discovery approach with LOGO does not seem to be the most effective method for mastering programming or problem solving skills [15, 20].

The amount of lesson structure and direct instruction provided by the teacher is the most obvious way to provide learning environments that can be hypothesized to result in observable and measureable differences in student performance.

The amount of instructional guidance and structure provided by the teacher can be placed on a continuum. The student discovery method lies at one end of such a continuum. As envisioned by Papert, students are instructed in a few basic LOGO commands and are then allowed to proceed at their own pace and through their own learning style to explore the LOGO microworld [3]. At the other extreme is an analytic, or top-down approach. The teacher provides the students with complete LOGO procedures, which the learner enters into the computer as is [21]. This teacher structured method is similar to traditional instructional situations since the emphasis is placed on product, as opposed to process. Research in other contexts has shown that such a highly structured approach to instruction does not foster student development of cognitive and metacognitive skills [20]. Between these two extremes is the teacher guided discovery approach. The teacher's role is to orchestrate and mediate the interaction between learner and computer [20, 21]. Emphasis is placed on processes--the process of LOGO programming as well as the cognitive processes the student needs to solve the problems presented.

Research Questions

The purpose of this study was to investigate the relationship between methods of teaching LOGO programming, learner characteristics, and performance on programming and mathematics tasks. Three research questions guided the study: a) What effect do learner characteristics (prior mathematics ability, cognitive style, gender, creativity, and self esteem) have on LOGO programming performance?; b) Do the interactions between learner characteristics and instructional methodology effect programming performance?; and c) Does method of LOGO programming instruction effect students' understanding of the concept of variable?

Method

Participants

One hundred five students (56 females, 49 males) participated in the project. The students were entering their first or second year of high school and ranged in age from fourteen to sixteen years. Student achievement level was determined by their scores on the mathematics subtests of the Comprehensive Test of Basic Skills (Level H Form U). Students ranged in ability from a grade level equivalent of 4.0 to 10.3. The LOGO project was part of an eight week program conducted during the summer of 1987 on the Louisiana State University campus. The major academic components of the program included classes in language arts, mathematics, and LOGO programming. The 105 participants were assigned randomly to one of eight LOGO class sections such that there were no more than fourteen students per section. Although some of the students had prior experience in programming, none the them had been exposed to the LOGO language. Four instructors were assigned to teach two class sections each.

The Treatment

Three approaches to teaching LOGO were selected for the project implementation. The first approach was teacher structured and was characterized by a top down, or analytical approach to teaching the language. In this approach, the teacher presented students with examples of procedures written in well-structured code. In addition, specific projects were assigned and students worked toward teacher-defined goals. The second approach utilized was the teacher guided method. This approach is characterized as bottom-up such that students are introduced to the primitives first and then given guidelines on how to combine the primitives into procedures. Activities start out as teacher directed, but later students are encouraged to identify their own problems and projects on which to apply newly learned concepts. The third approach was student discovery. In this approach, the teacher provided very few guidelines. Primitives and concepts were introduced, but students were given freedom to explore application of the primitives and concepts as they chose. In this approach, the instructor circulated about the lab and provided individual help as needed.

Curriculum

The Logo curriculum for the eight weeks included the use of the turtle graphics mode of the LOGO language. The MIT dialect of the language was used. The scope of the curriculum included use of sixteen primitives, the creation of procedures and editing commands needed to work with procedures, and some general programming concepts. The sixteen primitives included turtle manipulation commands (DRAW, FD, BK, RT, LT, PU, PD, HT, ST, REPEAT, SETBG, SETPC, HOME, CS, SETXY, SETH). The concepts included in the scope of the curriculum consisted of use of repeat, one variable procedures, recursion, and modularization.

Student Testing

Tests were administered to students to get measures of cognitive style, creativity, self-esteem, variable understanding, and programming performance. Since cognitive style is considered to be a stable construct at the high school level, one testing for cognitive style was administered. Two cognitive style tests were given: the Group Embedded Figures Test to determine field-independence-dependence and the Locus of Control test to determine whether students had internal or external centers of control. The Tennessee Self Concept Test was used as a measure of self esteem, and was administered both at the beginning and end of treatment. In addition, students took pre and post tests of understanding of the concept of variable. A test of programming performance was administered at the end of the eight week treatment. The programming test consisted of four subtests which measured students' ability to predict output, sequence sets of instructions to obtain desired output, use modularization to solve programming problems, and creativity. A total programming score was derived from the sum of the four subtest scores.

Procedures

Student instruction. The three treatments (teacher structured, teacher guided, student discovery) were assigned randomly to the eight sections such that each instructor had two different treatments to implement. After assignment of the treatments to the sections, there were three sections of teacher structured instruction, three sections of teacher guided instruction, and two sections of student discovery. All students received three hours of LOGO instruction per week for a period of eight weeks. The instruction took place in a computer laboratory that was equipped with 14 APPLE IIe computers with one disk drive and a color monitor.

Teacher training. The four instructors were certified in the area of mathematics and had prior experience in teaching programming. One instructor had prior experience in teaching LOGO. The instructors participated in twelve hours of in-service training conducted over a period of three weeks prior to the start of the project. The training sessions focused on teaching the programming concepts characteristic of the LOGO language and the language syntax. In addition, during the implementation of the project the teachers met with the researcher on a weekly basis to discuss student progress and questions related to use of the LOGO language.

Statistical Results

Do learner characteristics effect LOGO programming performance?

The data were analyzed using multiple regression procedures. A statistical model was created so that the relationship between student characteristics and programming performance could be examined. The student characteristics of interest included gender, cognitive style (field-independence-dependence), locus of control (internal or external), mathematics achievement (computation and concepts), creativity, and self esteem. The squared multiple R was .80 indicating these variables accounted for 80% of the variance in programming performance scores. The significant ($p<.05$) student characteristics included gender, prior mathematics computational skill achievement, and cognitive style. Males tended to outperform females, students with better computational skills achieved higher programming performance scores, and students with more field independent cognitive style performed better than those with field dependent cognitive style.

Do the interactions between learner characteristics and instructional methodology effect programming performance?

A multiple regression model was created to examine the significance of the interactions between each student characteristic and instructional method. Three interactions between student characteristics and instructional methodology were significant ($p<.05$) including the gender by treatment interaction (Table 1), self esteem by treatment interaction (Table 2), and the locus of control by treatment interaction (Table 3). As can be seen in Table 1, females tended to demonstrate better performance in the teacher structured classes, whereas males tended to do better in the teacher-guided classes. The lowest performance was observed for females in the student discovery classes. For the self esteem by method interaction, students with low self esteem performed significantly better in the teacher structured or guided classes. The lowest performance was observed for students with low self esteem in the student-based discovery classes. For locus of control, students who are more internally motivated perform equally well with any of the three methods used. However, students characterized by external locus of control performed significantly poorer in the student discovery classes.

Does method of LOGO programming instruction effect student understanding of the concept of variable?

A univariate analysis of variance was performed to test the effect of method on students' performance on the test of variable understanding. Method of instruction was a significant factor. Post hoc tests of differences between students in the three method groups revealed that the students in the teacher guided group (mean=11.4) performed significantly better than the students in the discovery group (mean=7.7).

Educational Implications

The major findings from this study indicate that student characteristics, method of instruction, and the interaction between these two factors influence student programming performance. When teacher structured, teacher guided, and student discovery methods of teaching programming are compared, it is clear that the teacher guided approach is more consistently found to be effective. This approach is more effective for both performance on the programming test as well as for transfer of skills to a test of variable understanding. In contrast, the student discovery approach was found to be the least effective, a finding similar to that of other researchers [20]. It is apparent that students must have both high levels of self esteem and more external locus of control to attain optimum programming achievement within the context of a student discovery based learning environment.

New information revealed by the current study is the significance of the interactions between certain learner characteristics and instructional methodology. It is obvious that these interactions are critical factors in the success or failure of certain students. Whereas students with more internal locus of control do equally well regardless of method, a troublesome difference in performance between the student discovery and teacher structured classes exists for students with external locus of control. A similar pattern is observed for students with low and high self esteem. Students with higher levels of self esteem perform equally well in all three treatment groups. However, students with lower self esteem exhibit the best performance in the teacher guided group and the poorest performance in the student discovery group.

Instructional methodology consists of the procedures and strategies used to bring about desired learning outcomes. The findings of the current research indicate that careful consideration should be given to the strategies utilized for teaching students with varying learner characteristics. Carefully chosen strategies of instruction can reduce the achievement disparities between students with differing characteristics rather than increase the differences between students. Educators should consider the psychological, emotional, and cognitive characteristics of students in the planning and implementation of computer programming instruction.

References

1. T. Shuell, Individual Differences: Changing Conceptions in Research and Practice, American Journal of Education, 94, pp. 357-377, 1986.

2. M. Chen, Children, New Media, and Microcomputers: Continuities of Research, in Children and Microcomputers: Research on the Newest Medium, M. Chen and W. Paisley (eds.), Sage Publications, Beverly Hills, 1985.

3. S. Papert, Mindstorms: Children, Computers, and Powerful Ideas, Basic Books, New York, 1980.

4. R. Taylor (ed.), The Computer in the School: Tutor, Tool, Tutee, Teachers College Press, New York, 1980.

5. R. Weisberg, Creativity: Genius and Other Myths, W.H. Freeman, New York, 1986.

6. D. Perkins, The Mind's Best Work, Harvard, Cambridge, Mass., 1981.

7. J. Dalbey and M. Linn, The Demands and Requirements of Computer Programming: A Literature Review, Journal of Educational Computing Research, 1, pp. 253-274, 1985.

8. S. Sharma, Learners' Cognitive Styles and Psychological Types as Intervening Variables Influencing Performance in Computer Science Courses, Journal of Educational Technology Systems, 15, pp. 391-399, 1986-87.

9. J. Pommersheim and F. Bell, Computer Programming Achievement, Cognitive Styles, and Cognitive Profiles, AEDS Journal, 20, pp. 51-59, 1986.

10. N. Webb, The Role of Gender in Computer Programming Learning Processes, Journal of Educational Computing Research, 1, pp. 441-458, 1985.

11. C.A. Bradley, The Relationship Between Students' Information-Processing Styles and LOGO Programming, Journal of Educational Computing Research, 1, pp. 427-433, 1985.

12. D.H. Clements, Delayed Effects of Computer Programming in LOGO on Mathematics and Cognitive Skills. Paper presented at the annual meeting of the AERA, San Francisco, CA., 1986.

13. J. Horton and K. Ryba, Assessing Learning with LOGO: A Pilot Study, The Computing Teacher, 14(1), pp. 24-28, 1986.

14. H.J. Becker, The Importance of a Methodology that Maximizes Falsifiability: Its Applicability to Research About LOGO, Educational Researcher, 16(5), pp. 11-16, 1987.

15. R. Pea, D.M. Kurland, and J. Hawkins, LOGO and the Development of Thinking Skills, in *Children and Microcomputers: Research on the Newest Medium*, M. Chen and W. Paisley (eds), Sage Publications, Beverly Hills, 1985.

16. D.H. Clements, Effects of LOGO and CAI Environments on Cognition and Creativity, *Journal of Educational Psychology*, *78*, pp. 309-318, 1986.

17. _____, Longitudinal Study of the Effects of LOGO Programming on Cognitive Abilities and Achievement, *Journal of Educational Computing Research*, *3*, pp. 73-94, 1987.

18. D.N. Perkins, The Fingertip Effect: How Information-Processing Technology Shapes Thinking, *Educational Researcher*, *14*(7), pp. 11-17, 1985.

19. B.K. Beyer, *Practical Strategies for the Teaching of Thinking*, Allyn and Bacon, Boston, 1987.

20. V.R. Delclos, J. Littlefield, and J.D. Bransford, Teaching Thinking Through LOGO: The Importance of Method, *Roeper Review*, *7*, pp. 153-156, 1985.

21. A. McDougall, Approaches to Teaching LOGO Programming, *Proceedings of the 4th World Conference on Computers in Education*, North-Holland, New York, 1985.

Table 1

Mean Programming Performance Score by Treatment and Gender

Treatment	Gender Male	 Female
Teacher Structured	23.0	21.6
Teacher Guided	34.0	20.4
Student Discovery	23.0	13.3

Table 2

Mean Programming Performance Score by Treatment and Self Esteem

Treatment	Self Esteem Low	 High
Teacher Structured	26.4	17.0
Teacher Guided	32.7	24.9
Student Discovery	15.2	17.7

Table 3

Mean Programming Performance Score by Treatment and Locus of Control

Treatment	Locus of Control Internal	 External
Teacher Structured	19.7	31.0
Teacher Guided	23.5	29.7
Student Discovery	18.7	10.3

AUTHOR INDEX

Adams, Elizabeth S. 273
Adams, Richard 146
Adams, William R. 253
Adler, Bill 165
Akalou, W. M. 20
Allmaker, Peter 304
Ambron, Sue Ann 6
Anderson, Patricia 113
Anderson, Ronald E. 31, 118
Anderson, W. Steve 281
Anderson-Inman, Lynne 165
Ashby, Bill 362
Atkins, Bill 162
Ayers, Sharon Whitton 141
Bantz, David 3
Barlow, Judith A. 357
Baskerville, Richard L. 237
Bateman, Kitty 353
Batoff. Mitchell E. 173
Bauer, Charles R. 278
Becker, Henry Jay 136, 285
Becker, Kim 353
Bellamy, Shirley H. 252
Benoit, Eugene 153
Berger, Carl 115, 195
Bethke, Doris "Dee" 139
Beverly, Don 181
Biediger, Carolyn V. 113
Biglan, Barbara 246
Billings, Karen 115, 168
Bingham, Margaret 131
Bitter, Gary 141
Blankenbaker, Ruthie 141
Blaschke, Charles L. 31
Bolter, Jay 3
Boone, Randy 171
Bork, Alfred 292
Bowman, Sally 351
Breit, Frank 162
Breslawski, Stephen 245
Brett, George 3
Brisk, Suzanne 357
Brittain, David 140
Brown, Barbara 148
Brown, David W. 322
Brown, Susan 22
Bruce, Bertram 80
Budin, Howard 7, 115, 170
Bullock, Donald W. 215
Burney, James D. 249
Burrows, David 162
Butler, Greg 186
Butzin, Sally 190
Caldwell, Robert M. 209, 362
Campbell, J. 131
Card, Barbara 153
Carey, Doris 132
Carey, Ernest 110
Carlisle, James 25
Carlson, Elizabeth A. 115
Carnow, Gary A. 239
Carter, Richard 325, 332
Carver, Vivien 287
Catalano, Ruth 181
Cemen, Pamela Byrd 120
Charischak, Ihor 2
Chilson, David Warner 35
Cinque, Harriet A. 49
Clark, G. Christopher 34
Collis, Betty 285, 349
Colman, Anne 132
Conery, Leslie 343
Connally, Frank 168
Cook, John E. 173
Cory, Sheila 141, 170, 349
Cottingham, Esther 190
Cowan, Hilary 239
Cron, Mary 165
Crummett 110
Cunniff, Nancy 7
D'Amico, Thomas 173
Davis, Frank 325
Dede, Christopher 95
DeLauder, William B. 248
DeLoatch, Sandra J. 248
Denning, C. 131
Dews, Shelly 245
Dodl, Norman R. 308
Donham, Jean 144
Edwards, Carol 196
Elliott, Alison 249
Ellis, James D. 346
Ellis, Monica 353
Engen-Wedin, Nancy 281
Epstein, Kenneth 287
Evans-Huse, Vanessa 322
Feddern, Barbara 141
Ferrari, Frank J. 110, 176
Feurzeig, Wallace 325, 332
Finkel, LeRoy 31
Fisher, John C. 281
Fleming, Elyse 72
Ford, Mary Jane 239
Foster, John L. 230
Foster, Leon 153
Franklin, Betsy 362
Fredman, Alice 118
French, Michael P. 192
Friel, Susan 80
Friske, Joyce S. 169
Frye, Douglas 224
Furlong, Mary 6, 362
Gaddis, William 196
Gaede, Owen F. 166
Gehring, Philip F. 197
Gering, Dawn 173
Geyer, Pete 362
Giangrande Jr., Ernie 304
Gleason, B.J. 230
Gonzales, Andrea 196
Gonzales-Baker, Varia 165
Goodson, Bobby 140
Gordon, Paul 113
Graeber, Janet 362
Grantham, Darlene 28, 77
Greenberg, Gary 192
Groninger, Cheryl K. 137
Gwyn, Rhys 188
Hankins, Dale 245
Hardy, Cliff 75
Harris, Joseph R. 252
Harrison, George C. 248
Hart, Joyce L. 162
Hartig, Heidi E. 230
Hartle, Helen 132
Hartman, Carol 153
Hasloff, Gary 34
Hathaway, David L. 137
Hayward, Lee 25
Heebner, Amy 115
Heidt, Dede 141
Hein, Kristen 353
Heller, Rachelle S. 56
Hewitt, Louise C. 278
Higgins, Kyle 171
Higgins, Norman 113
Hillberry, Joseph B. 190
Hillinger, Michael 165
Hiltz, Starr Roxanne 230
Hinely, Reg 162
Hoelscher, Karen J. 355
Holloway, M. 131
Holtznagel, Don 139
House, Beverly 153
Hunter, Beverly 6, 82
Ice, Jerry 243
Imhof, Heidi 146
Jacobi, Carolyn 132
Jacobs, Judith E. 115
Janssen, Betsy 242
Johnson, Dennis A. 181
Johnson, Jerry 120
Johnson, Vivian 343
Jones, Robert 3
Jongejan, Tony 80, 140
Kahn, Ted M. 119, 360
Keifer, Steven P. 110
Kendall-Piel, Colleen 278

Kerr, Ruth 239
Kertzman, Wanda 28
Kinnaman, Pam 183
Kirkpatrick, Susan N. 246
Knezek, Don 140
Knight, Anne H. 31
Knight, Martha 283
Knight, Roger 362
Knouse, Jean 176
Kornecki, Andrzej 237
Kosloski, Barbara C. 20
Kovatch, Nancy 2
Kravitz, Edith 351
Kruidenier, John 28
Land, Michael 166
Langhorne, Mary Jo 144
Latarski, Don 344
Lawson, John 141, 170, 195
Leonard, Jenelle V. 252
Leventhal, Laura Marie 35
Lewis, Ed 237
Lewis, Jesse C. 248
Liden, Larry E. 28
Ligon, Helen H. 289
Lintner, Mildred D. 132, 169, 243
Little, Roberta Sloan 166
Lloyd, Jackie L. 346
Locketz, Irving 173
Lockwood, Lamont 209
Lough, Tom 2, 168
Lundberg, Priscilla 181
Lynch, David 153
Makings, David 243
Malbasa, Veljko D. 263
Manwaring, Mark L. 263
March, Dennis 283
Marciano, Vincent 141
Marcus, Stephen 4
Marsh, Mary 242
Marshall, Gail 349
Martin, C. Dianne 56
Martorella, Peter H. 162
Matheson, Keborah 110
Mathies, Bonnie 169
Maury, Kathleen 75
McAvoy, Constance 171
McCarthy, Regan 360
McClintock, David N. 159
McCormick, Ann H. 119, 360
McCoy, Leah P. 308
McDonald, Gary 289
McDonald, Merry 289
McGrath, Diane 322
McGraw, Karen 148
McGrew, Robin 146
McLaurin, James L. 2
McMillan, Marlene 118
Merritt, Susan 75
Mlynarczyk, Charles 169
Moonen, Jef 188, 285
Moore, Pamela 169
Morgan, James M. 159
Morrell, Joseph S. 357
Morrell, Judith H. 301
Morrison, Virginia 32
Morton, Chris 181
Moursund, David 115, 170, 195, 343
Murphy, Ellen R. 77
Myers, Cynthia A. 287
Neighbours, David 289
Niguidula, David 28
Noel, Michael L. 357
Noqueira, Flynn 197
Norton, Cheryl J. 357
O'Brien, Thomas C. 119
Odom, L. Joy 77
Ohme, Paul 283
Olds, Henry 27
Olson, David 278
Olson, Richard 165
Ortiz, Albert 32
Osterling, Phillip 153
Owens, Virginia L. 243
Parker, Janet 22
Pearson-Jagoe, Katherine 246
Perelgut, Stephen 110
Perkins, D. N. 28
Persson, Marilyn P. 278
Peyton, Marvin 183
Pitts, Anne W. 249
Plourde, Paul J. 173
Poage, Julie 186
Poe, Virginia 239
Poirot, Jim 141
Ponder, Gerald 162
Pooch, U. W. 293
Pope, Jack W. 113
Pratt, Donald L. 183, 354
Prewitt, Gray 354
Price, Bonnie 183
Price, Ricki S. 77
Price, Robert V. 246
Prohask, Bonnie 165
Radabaugh, Mary Pat 196
Raker, Brenda M. 22
Randolfi, Ernesto A. 346, 357
Rawitsch, Don 348
Rawitsch, Peter 348
Ray, John R. 183
Rehmke, Denise 144
Richards, John 332
Richman, Ellen 118
Roberts, Linda G. 31
Roberts, Lon 113
Roberts, Nancy 325, 332
Rodabaugh, Karl 287
Rogers, Rodney O. 237
Romance, Nancy 346
Rosen, Marion 2, 22
Roth, R Waldo 291
Rouse, Bill 135
Ryder, Bernice 77
Sabin, Roberta Evans 281
Sampson, J. 293
Sanchez, Nancy 28
Sanders, Jo 118, 196
Sangston, Beverly J. 28
Santomauro, Dennis J. 49
Sassi, Anthony 22
Saterfiel, Thomas H. 139
Schartz, Lenny 362
Scheftic, Carol 312
Scherer, Charlotte 287
Schmidt, Denise A. 32
Schmitt, Christy 20
Schmitz, Valerie 173
Schreihofer, Ellen S. 230
Schultz, Andy 289
Schwartz, Helga 353
Schwartz, Steven H. 28
Seamans, David A. 263
Seaton, William J. 243
Segal, Gerry 141
Sherrill-Range, Tina 183
Shirazi, B. 293
Shore, Ann 337
Shuller, Stephen M. 190
Silverman, Stan 132
Singletary, Carol H. 190
Slater, Donald J. 20
Smaldino, Sharon E. 183
Smith, Linda G. 32
Smith, Richard Alan 22, 77
Soloway, Elliot 224
Sparks, JoAnne 357
Spielvogel, Robert 115
Stedman, James B. 31
Stokes, Sue 113
Stone, Antonia 196, 355
Strezeski, Mike 139
Strudler, Neal 186
Stumvoll, Linda K. 192
Sutton, Mary 289
Sutton, Rosemary 72
Talburt, John R. 281
Tanik, M. M. 293
Taylor, Ann 353
Taylor, John T. 137
Taylor, Robert 115, 170, 195
Thomas, Lajeane G. 13, 169
Tigani, Grace 77
Tobin, Dianne 192
Tu, Jho-Jo 322

Tubb, Gary W. 162, 237
Tuscher, Leroy J. 362
Ullian, John A. 357
Upton, Mary 283
van Deusen, Robert M. 144
Varma, Indu 159
Vitale, Michael R. 346
Walker, Decker 285
Walker, Michele 249
Wallace, Susan R. 45
Walters, Leigh 34
Ware, Linda 77
Warner, Michael P. 181
Watkins, Nellouise D. 248
Watt, Daniel H. 27, 62, 141
Watt, Molly 27, 62, 141, 195
Wehrs, William 173
Weiner, Cheryl 168
Welch, Francis C. 115
Wholeben, Brent Edward 103
Wiburg, Karin 355
Wiebe, James 169
Wilson, Maria 355
Wintz, Cary D. 20
Woods, Brooke P. 249
Woodward, John 146
Wydra, Dennis 34
Yates, Billy C. 343
Yaverbaum, Gayle 245
Yoder, Sharon Burrowes 72, 127
Young, Jon 75
Zauderer, Marvin M. 360
Ziegler, Reinhard 245
Zigalo, Barbara 289